ASSIGNMENT 1 FRONT SHEET

Qualification	**BTEC Level 5 HND Diploma in Computing**		
Unit number and title	Unit 30: Application Development		
Submission date		**Date Received 1st submission**	
Re-submission Date		**Date Received 2nd submission**	
Student Name		**Student ID**	
Class		**Assessor name**	

Student declaration

I certify that the assignment submission is entirely my own work and I fully understand the consequences of plagiarism. I understand that making a false declaration is a form of malpractice.

Student's signature	

Grading grid

P1	P2	P3	M1	M2	D1

❐ **Summative Feedback:**		❐ **Resubmission Feedback:**
Grade:	**Assessor Signature:**	**Date:**
Lecturer Signature:		

SOFTWARE REQUIREMENTS SPECIFICATION

for

Training Management System

Version 1.0

Table of Contents

1. Introduction

1.1. PURPOSE

Training Management System 1.0 is a new-born system to satisfy FPT Co. desires to build the continuing study environment throughout the corporation by permitting users to manage the activity of “Training” for internal training program of the company, including:

- Manage trainee accounts
- Manage trainers
- Manage course categories
- Manage courses
- Manage topics
- Assign topic to course
- Assign trainer to topic.

1.2. ASSIGN TRAINEE TO COURSEDOCUMENT CONVENTIONS

No.	Abbreviations/Terms	Explanation
1	SRS	Software Requirement Specification
2	MW	Mockup wireframe
3	UC	Use Case
4	ERD	Entity Relationship Diagram
5	DFD	Data Flow Diagram
6	CRUD	Create-Read-Update-Delete a record in database
7	MVC	Model-View-Controller

TABLE 1. DOCUMENT CONVENTIONS TABLE

1.3. REFERENCES

No.	Name	Description

TABLE 2. DOCUMENT REFERENCES TABLE

2. HIGH LEVEL REQUIREMENT:

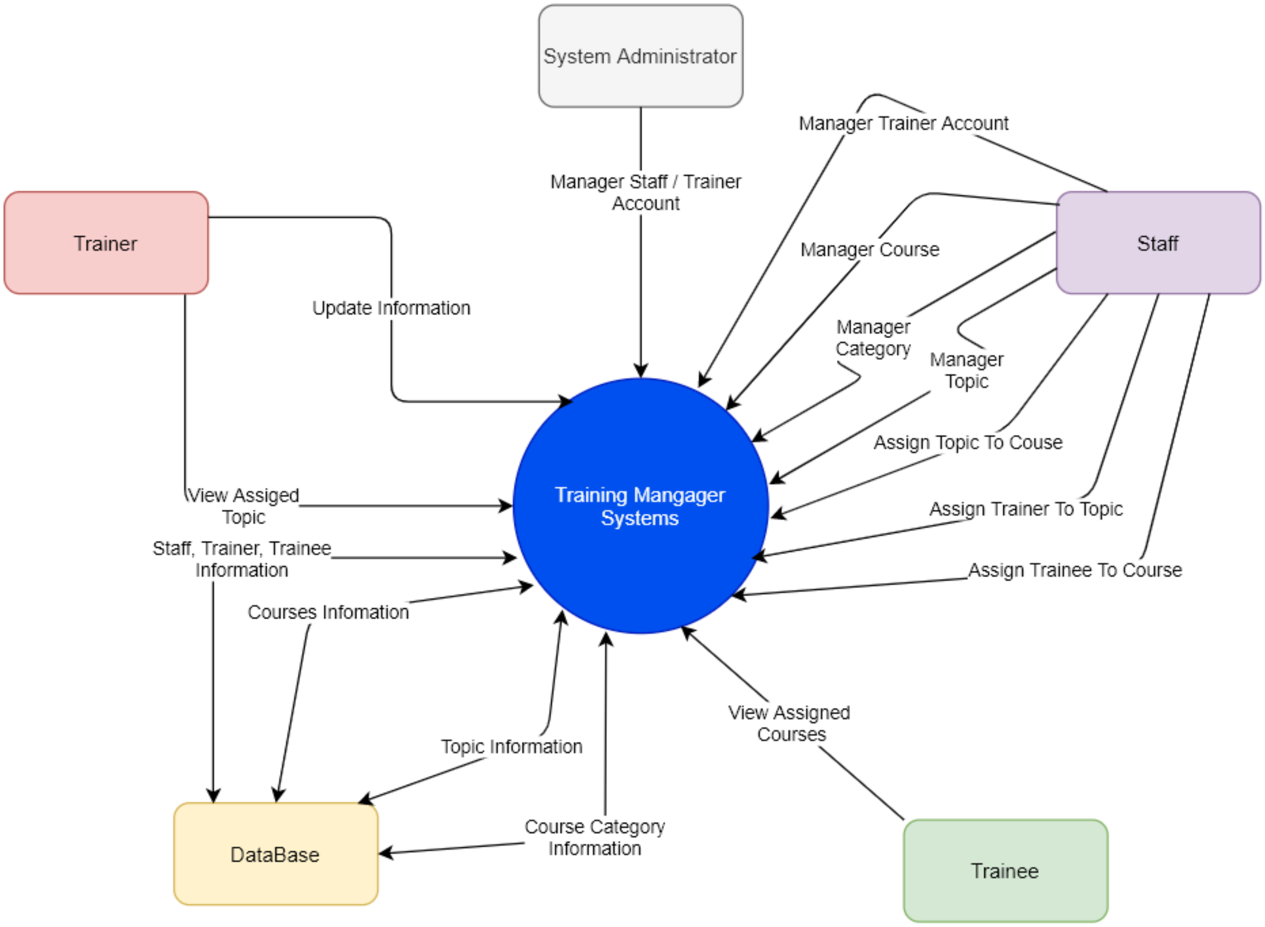
2.1. PRODUCT PERSPECTIVE:

Training Management System 1.0 is a newly launched system that meets University Of Greenwich desire to build a continuous learning environment throughout the company by allowing users to manage the “Training” activity for Company internal training programs, including:

- Manage trainee accounts
- Manage trainers
- Manage course categories
- Manage courses
- Manage topics
- Assign topic to course

- Assign trainer to topic
- Assign trainee to course

The context diagram in Figure 1 illustrates the external entities and system interfaces:



2.2.ACTOR AND USE CASE DESCRIPTION:

2.1.1. USE CASE DIAGRAM

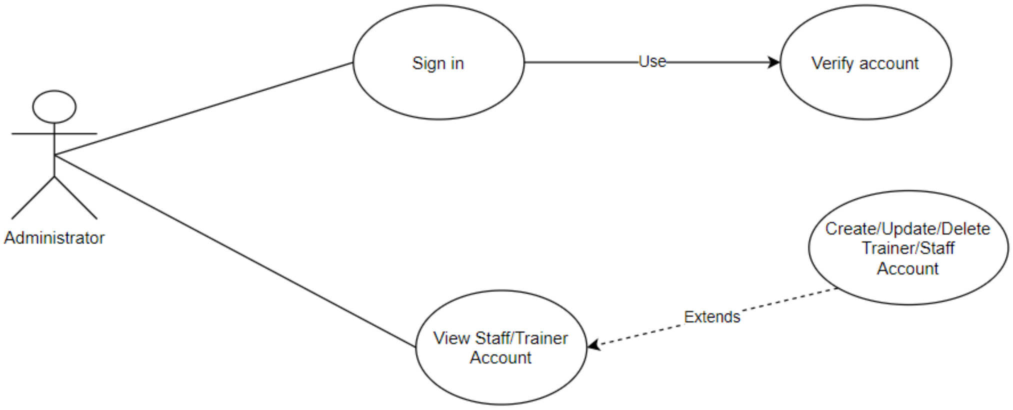

Sign in
Use
Verify account
Administrator
Create/Update/Delete Trainer/Staff Account
Extends
View Staff/Trainer Account

FIGURE 2. SYSTEM USE CASE DIAGRAM (ADMIN)

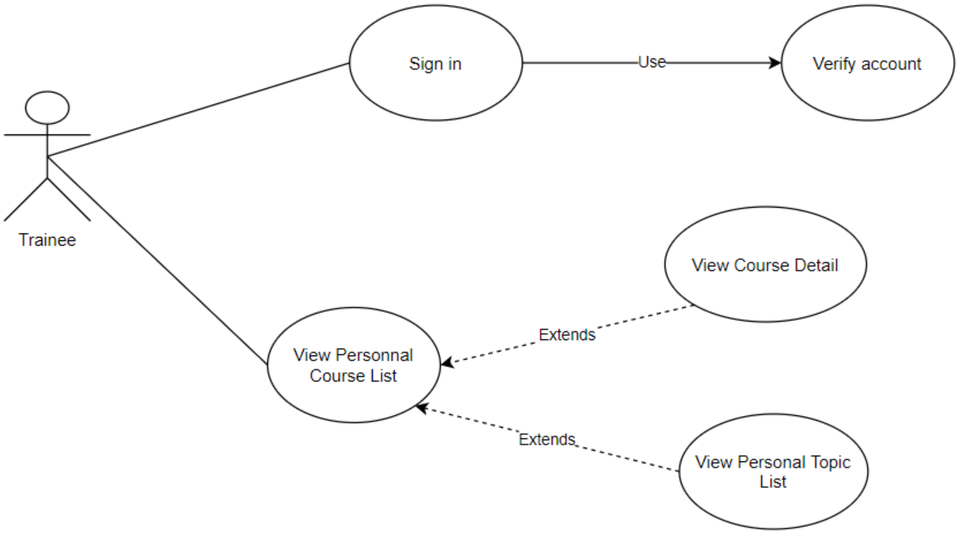


FIGURE 3. SYSTEM USE CASE DIAGRAM (TRAINEE)

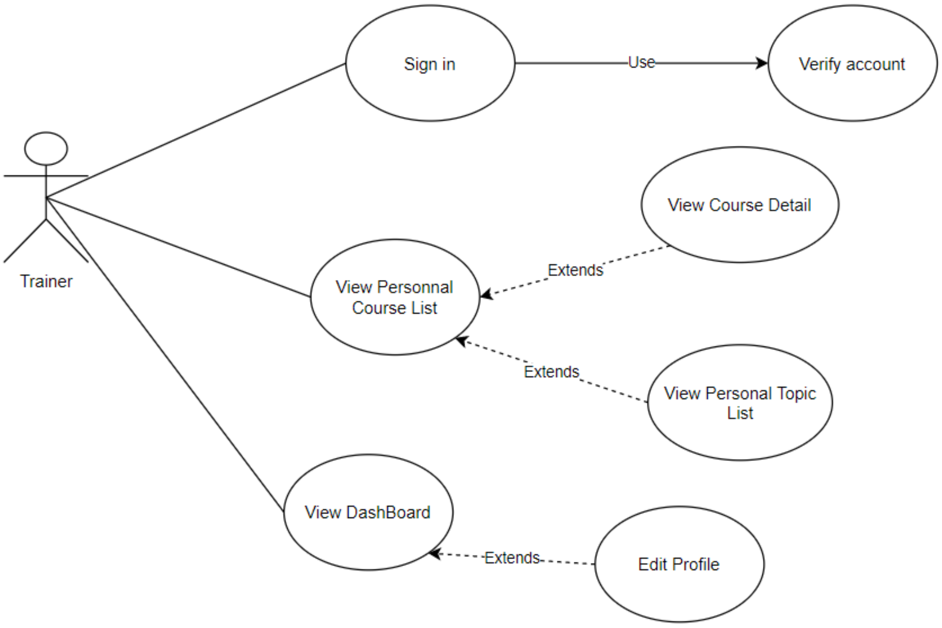


FIGURE 4. SYSTEM USE CASE DIAGRAM (TRAINER)

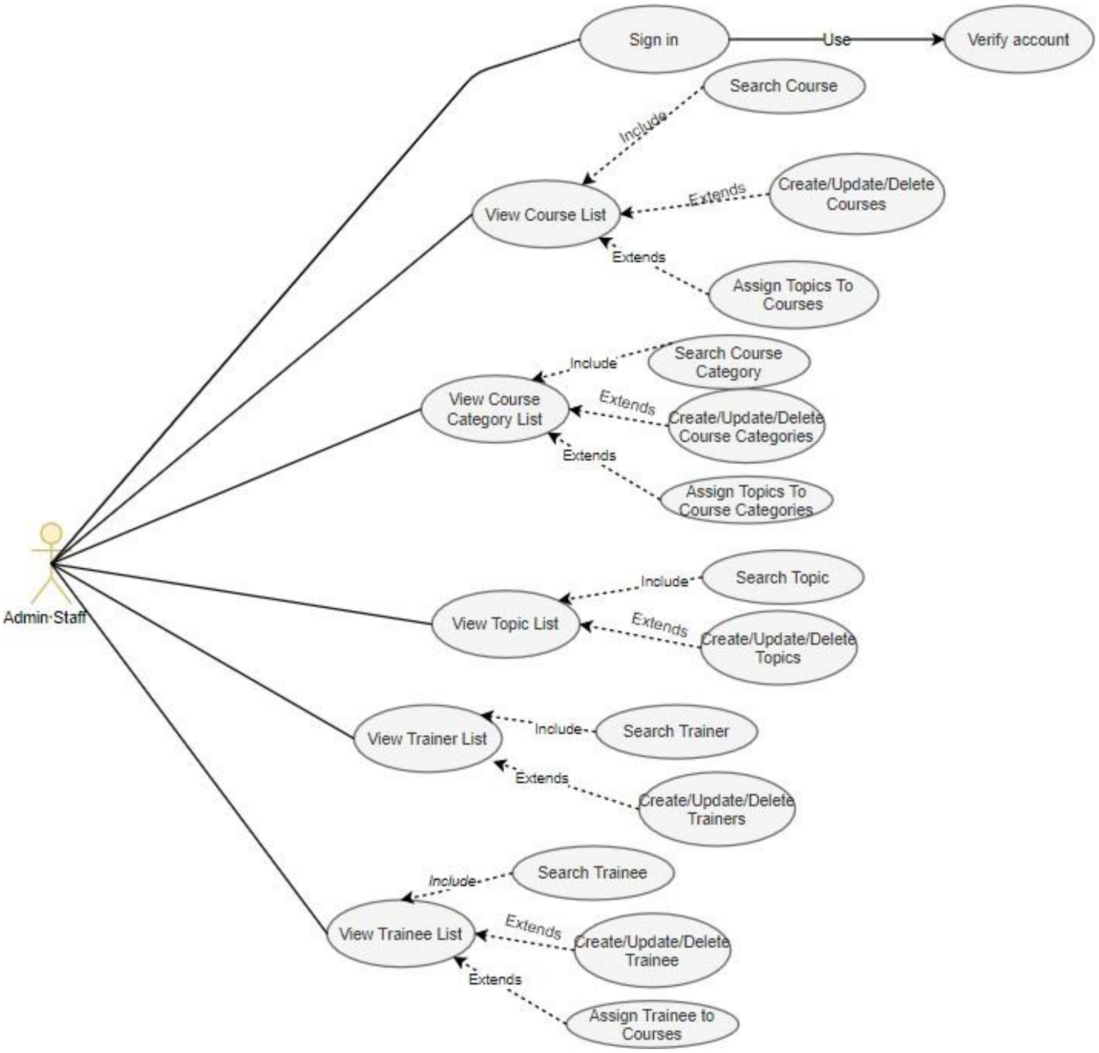


FIGURE 5. SYSTEM USE CASE DIAGRAM (TRAINING STAFF)

2.1.2. ERD DIAGRAM

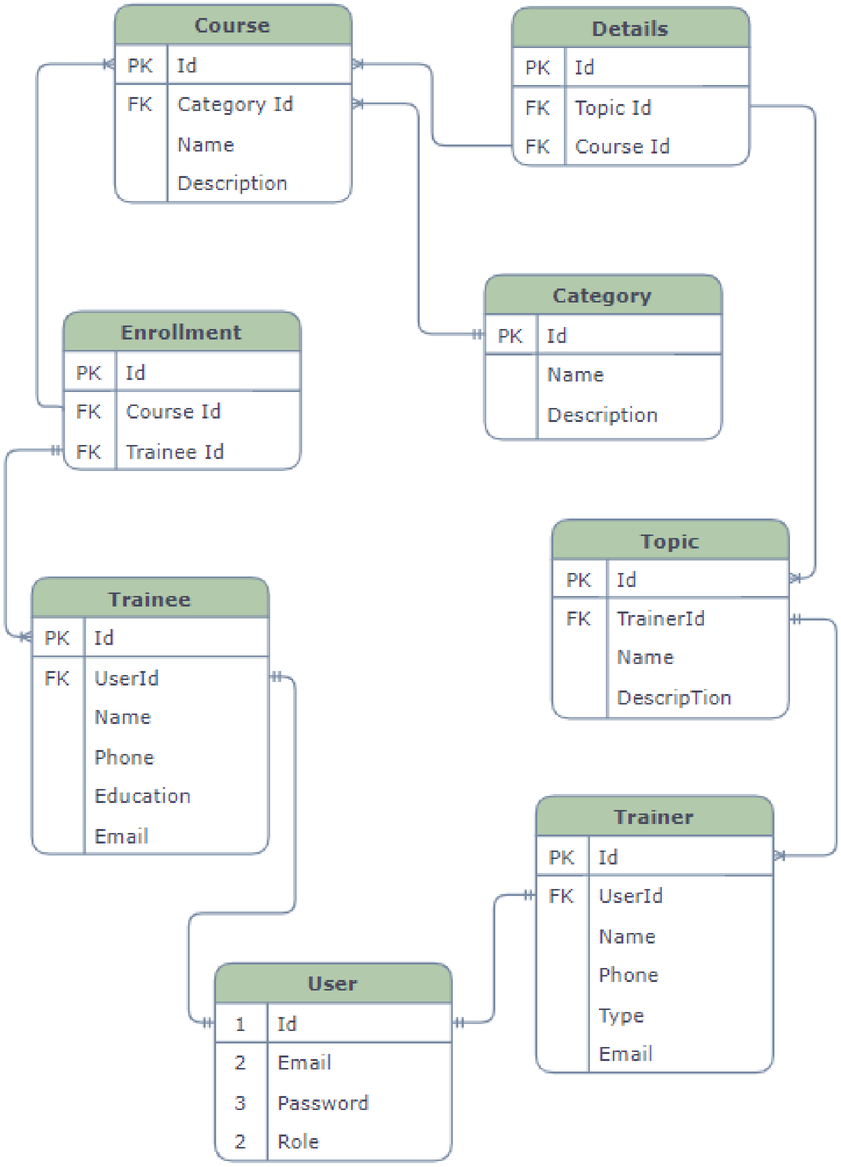


FIGURE 6. SYSTEM ENTITY RELATIONSHIP DIAGRAM

2.1.3. DFD DIAGRAMS

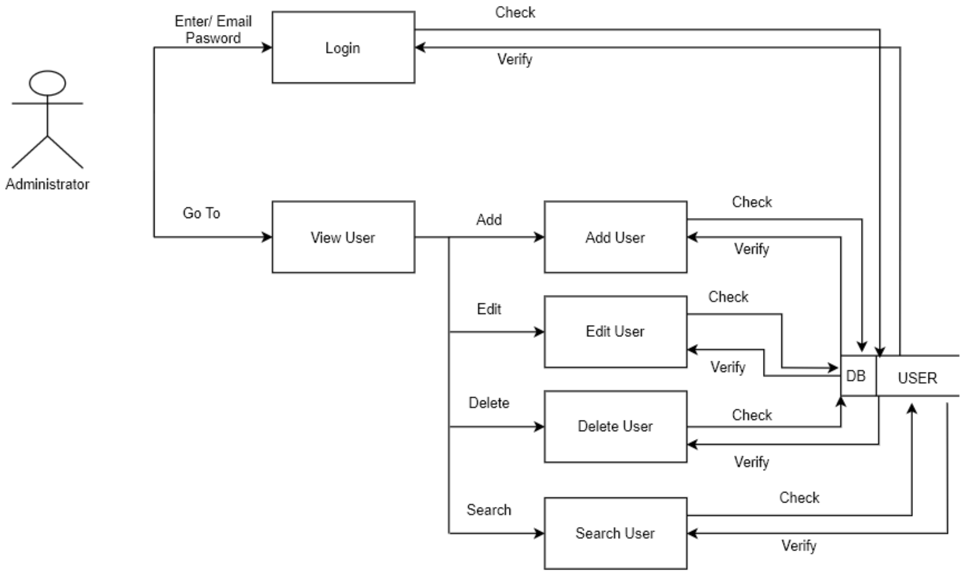


FIGURE 7. SYSTEM ADMINISTRATOR DATA FLOW DIAGRAM

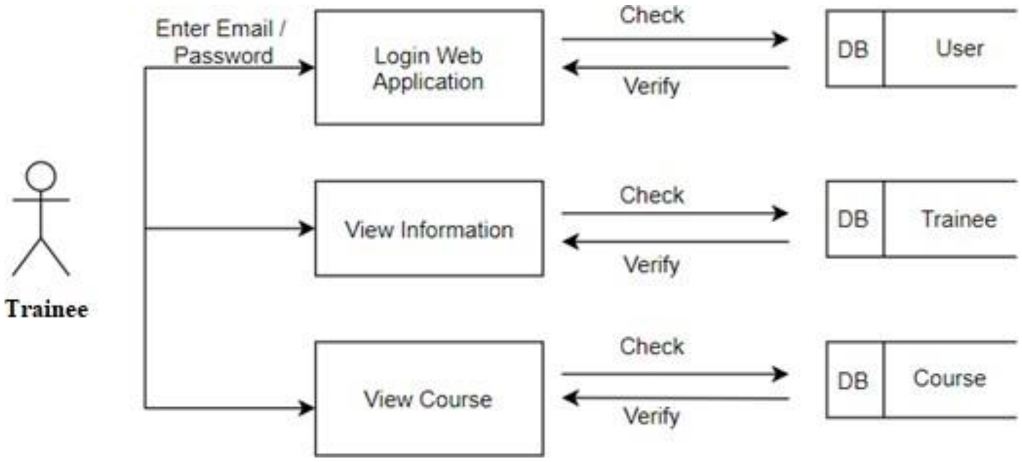


FIGURE 8. TRAINEE DATA FLOW DIAGRAM

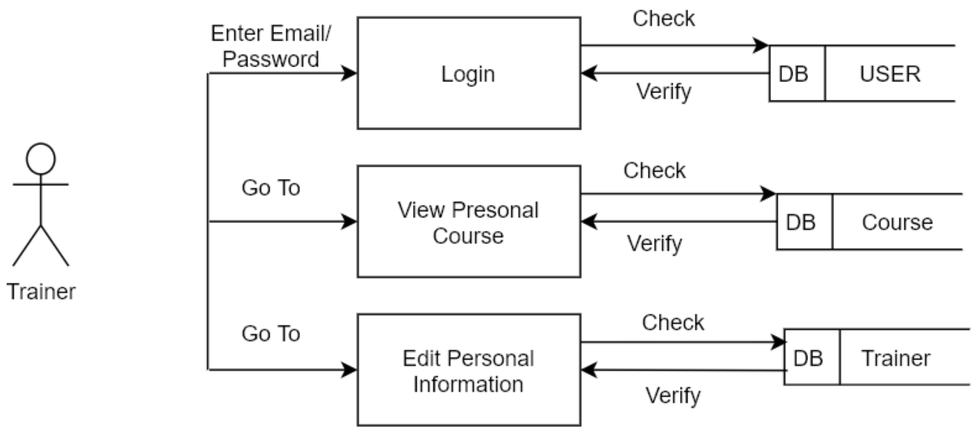


FIGURE 9. TRAINER DATA FLOW DIAGRAM

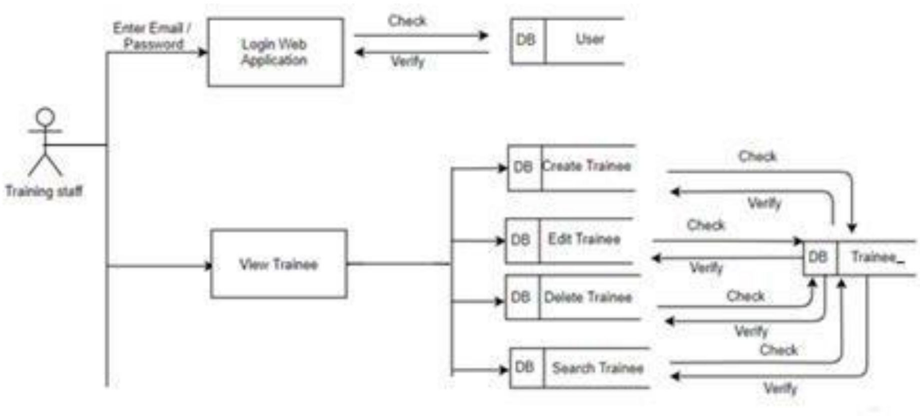


FIGURE 10. TRAINING STAFF DATA FLOW DIAGRAM 1

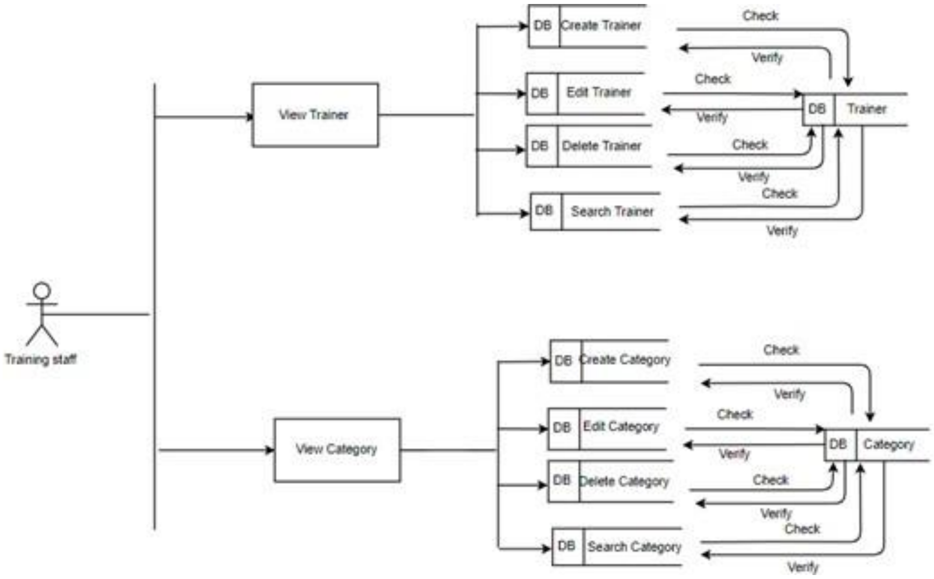


FIGURE 11. TRAINING STAFF DATA FLOW DIAGRAM 2.

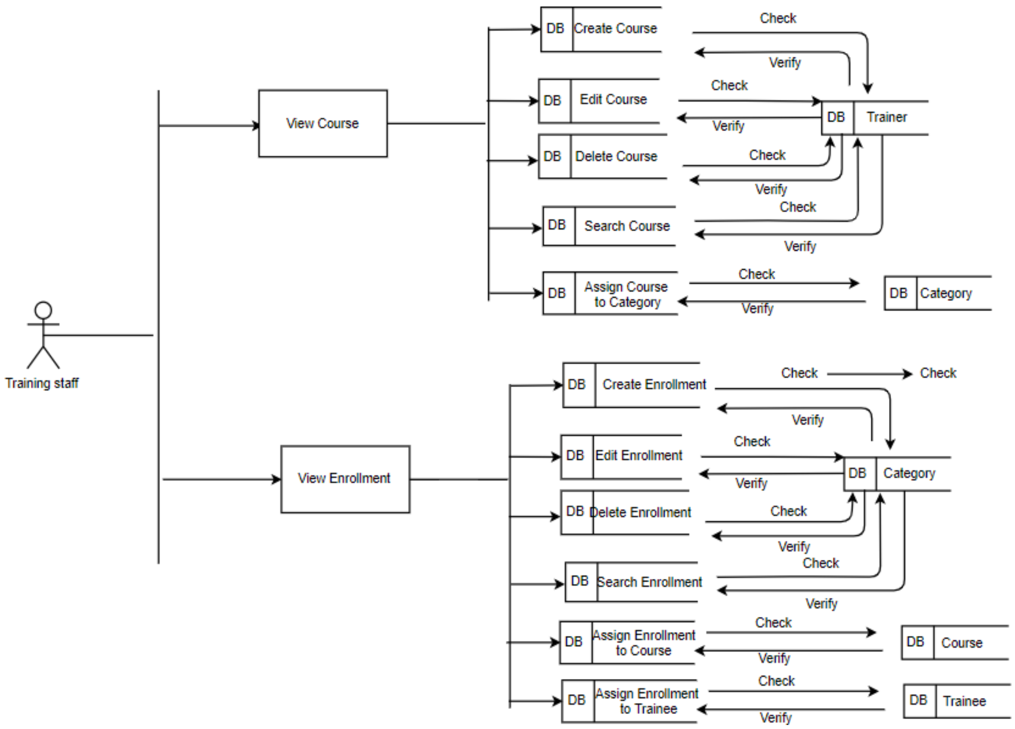


FIGURE 12. TRAINING STAFF DATA FLOW DIAGRAM 3.

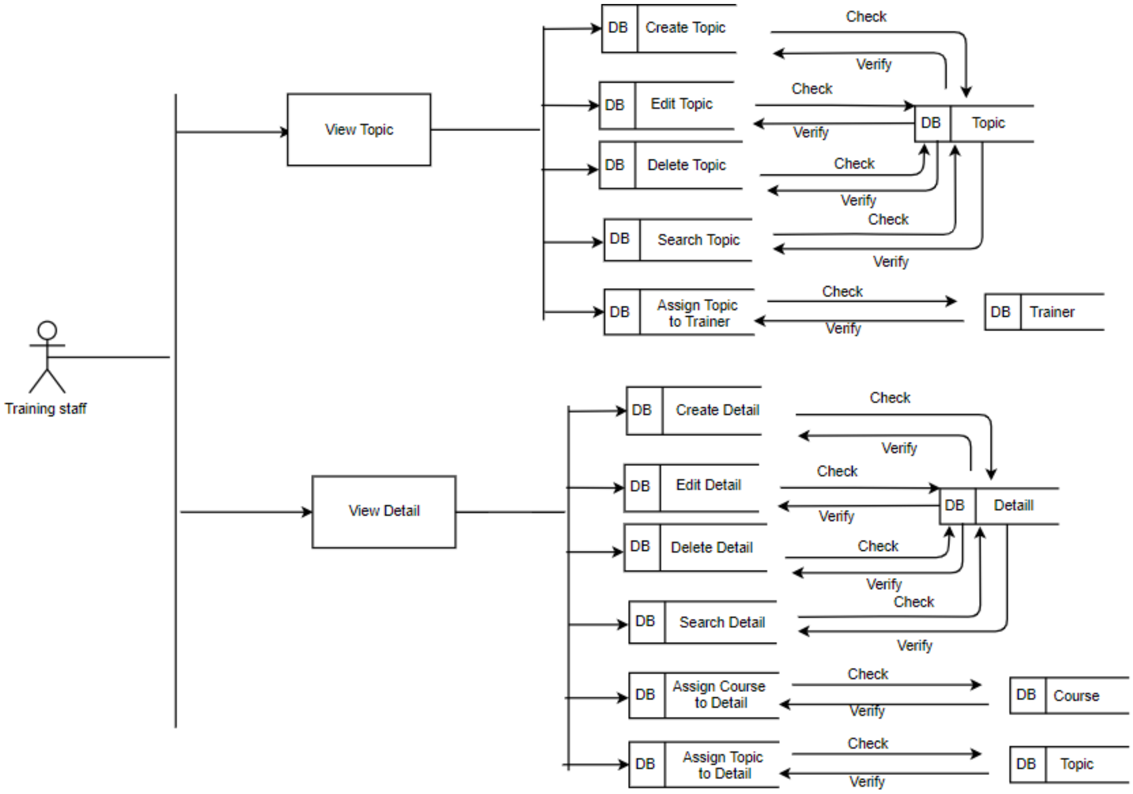


FIGURE 12. TRAINING STAFF DATA FLOW DIAGRAM 4.

2.1.4. ACTORS DESCRIPTION:

No.	Actor Name	Description
1	Administrator	The admin full system control
2	Training staff	The person authorized by the administrator to manage courses, course categories, guides, students and topics
3	Trainer	Actors coaching students through topics assigned to them
4	Trainee	The actors are assigned to attend courses and are trained by teachers

TABLE 3. ACTOR DESCRIPTION TABLE

2.1.5. USE CASE DESCRIPTION:

No.	Use Case ID	Use Case Name	Description
1	UC_LOGIN	Login system	User logged into the system
2	UC_AD_01	View account	Admin see all created accounts
3	UC_AD_02	Create a new account	Admin creates new user accounts for trainers/trainers
4	UC_AD_03	Edit existing account	Admin modifies user accounts
5	UC_AD_04	Delete existing account	Admin deletes account users
6	UC_TRER_01	View courses	interns see all courses specified
7	UC_TRER_01	View courses	Lecturer see all courses with topics assigned
8	UC_TRER_02	Update personal info	Trainer profile updates his / her
9	UC_STF_CO_01	View courses	staff trained to see all courses
10	UC_STF_CO_02	Create new	Staff train a new course
11	UC_STF_CO_03	course Edit existing courses	Training staff edit an existing course
12	UC_STF_CO_04	Delete an existing course	Training staff delete an existing course
13	UC_STF_CO_05	Add subject to a course	Trainer adds a subject to the course
14	UC_STF_CC_01	View course category	Training staff view all course categories
15	UC_STF_CC_02	Create a new course category	Training staff create a new course category
16	UC_STF_CC_03	Edit an existing course category	Training staff to edit existing course categories
17	UC_STF_CC_04	Remove existing Course categories	Training staff to delete existing course categories
18	UC_STF_TP_01	View the topic	Training staff See all topics
19	UC_STF_TP_02	Create a new theme	Training staff Create new topics
20	UC_STF_TP_03	Edit an existing theme	Training staff edit an existing theme
21	UC_STF_TP_04	Remove an existing theme	Training staff deletes an existing topic
22	UC_STF_TR_01	View Trainers	Training staff members to see all teachers
23	UC_STF_TR_02	Create a new Trainers	Training staff to create new trainers
24	UC_STF_TR_03	Edit existing Trainers	An existing faculty editing training staff
25	UC_STF_TR_04	Remove existing trainers	Training staff to delete an existing teacher
26	UC_STF_TE_01	View Internship	Training staff to see all students
27	UC_STF_TE_02	Create a new Trainee	Training staff to create new trainers
28	UC_STF_TE_03	Edit an existing Trainee	Existing student editing training staff
29	UC_STF_TE_04	Delete an existing Trainee	Training staff to delete an existing trainer
30	UC_STF_TE_05	Assign Trainees to participate in the course (the)	The training staff members to assign participants to the course (the)

TABLE 4. USE CASE DESCRIPTION

2.1.6 USE CASE AND ACTOR MAPPING:

Actor Use case	Administrator	Training staff	Trainer	Trainee
UC_LOGIN	V	V	V	V
UC_AD_01	V			
UC_AD_02	V			
UC_AD_03	V			
UC_AD_04	V			
UC_TREE_01				V
UC_TRER_01			V	
UC_TRER_02			V	
UC_STF_CO_01		V		
UC_STF_CO_02		V		
UC_STF_CO_03		V		
UC_STF_CC_04		V		
UC_STF_CC_05		V		
UC_STF_TP_01		V		
UC_STF_TP_02		V		
UC_STF_TP_03		V		
UC_STF_TP_04		V		
UC_STF_TP_05		V		
UC_STF_TR_01		V		
UC_STF_TR_02		V		
UC_STF_TR_03		V		
UC_STF_TR_04		V		
UC_STF_TR_05		V		
UC_STF_TE_01		V		
UC_STF_TE_02		V		
UC_STF_TE_03		V		
UC_STF_TE_04		V		
UC_STF_TE_05		V		

2.3.DEVELOPMENT ENVIRONMENT

	Developing (client) system	**Server system**
Operating system	Windows 10 Pro workstation	Azure Server
Processor	Intel Core i7 7700HQ	2x Intel Xeon E5-2690 - 2.90GHz x 8 Core
Storage	256GB SSD Samsung EVO NVME	4 TB SSD Samsung EVO
Memory	16 GB DDR4 Bus 24000mHz	12x16GB DDR3

TABLE 6. DEVELOPMENT ENVIRONMENT

3. FUNCTIONAL REQUIREMENT:

3.1.UC_LOG: LOGIN

Use Case Description

Name	Login	**ID**	UC_LOG
Description	Allow user to login to the system through the index page		

Actor	Administrator Staff Trainer Trainee	**Trigger**	User clicks the login button
Pre-condition	The login page loads completely		
Post condition	Account inputted are valid		

Activities

Actor		**System**	
Main flow: Login successfully			
1	The user inserts Username and Password into the appropriate text box and clicks the login button		
		2	Check the user name and password validation
		3	Check whether the role is an administrator
		4	Check the user name assignment process
		5	Move to default page based on user role

Business Rules

Rule no.	**Rule**	**Description**

System message

Message no.	**Message**	**Case**
1	"Username / Password incorrect. Please check again! "	Username / Password incorrect
2	"Username / Password does not exist. Please ask your administrator to create a new account! "	The account does not exist in the database
3	" Login is successful!	Exact Username and password

3.2.UC_AD_01: VIEW ACCOUNT LIST:

Use Case Description

Name	View account list	**ID**	UC_AD_01
Description	Allow the admin to view the system's account list		
Actor	Administrator	**Trigger**	Admin logs into the system Admin clicks the "Dashboard" button
Pre-condition	The user logged in with correct admin account		
Post condition			

Activities

Actor		System	
Main flow: View successfully			
1	Admin click the “Dashboard” button		
		2	Move to the “Account Index” page

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case

3.3.UC_AD_02: CREATE NEW USER ACCOUNT:

Use Case Description

Name	Create new user	**ID**	UC_AD_02
Description	Allow admin to create new Username/Password/Role for new user		
Actor	Administrator	**Trigger**	Admin clicks the “Create new” button
Pre-condition	The Account Index page loads completely		
Post condition			

Activities

Actor		System	
Main flow: Login successfully			
1	Admin click the button “Create new”		
		2	Move to the page "Create new"
3	Admin insert the Username / Password / Role of the new user they want to create	3	
		4	Verify User Name and Password
		5	Check for redundancy of Username
6	Admin click the “Create		
		7	Create a new version of account in the database using Admin's inserted data Member
		8	Print successful message

Business Rules

Rule no.	Rule	Description
1	Username is the version only	Every account needs a unique username

System message

Message no.	Message	Case
1	“Username / Password incorrect. Please check again! ”	Wrong / Invalid
2	“Username is taken! Please choose a different username "	Username / Password
3	" Please choose your account role! "	Username has already been used and is not allowed to be copied
4	"Account created successfully!"	Admin forgot to set the role for the account

3.4. UC_AD_03: EDIT EXISTING USER ACCOUNT:

Use Case Description

Name	Edit existing account	**ID**	UC_AD_03
Description	Allow user to edit Username/Password/Role of an existing account		
Actor	Administrator	**Trigger**	Admin clicks the “Edit” button
Pre-condition	The Account Index page loads completely and accounts list exist		
Post condition			

Activities

Actor		System	
Main flow: Edited successfully			
1	*Administrator clicks the button "Edit*		
		2	*Move to the page "Edit" with user ID*
3	Admin inserts New Username / Password / Roles		
		4	Checking verification of username and password
		5	Check for excess of the user name
6	Administrator click the "Save		
		7	Update the account version with an ID in the database by using the inserted data from the administrator.

Business Rules

Rule no.	Rule	Description

1	Usernames are unique instances	Every account needs to have a unique username

System message

Message no.	**Message**	**Case**
1	"Username / Password incorrect. Please check again! "	Wrong / not valid User Name
2	"Username is taken! Please choose a different username "	The user name is used and is not permitted to copy.
3	" Please choose your account role! "	The administrator forgot to play the role for the account
4	"Account has been updated successfully!"	Update the successful account and are stored in the database.

3.5.UC_AD_04: DELETE EXISTING USER ACCOUNT:

Use Case Description

Name	Delete existing account	**ID**	UC_AD_04
Description	Allow admin to delete existing account		
Actor	Administrator	**Trigger**	User clicks the delete button
Pre-condition	The Account Index page loads completely and the list of accounts exists.		
Post condition	The administrator presses "Yes" after a dialogue box appears to confirm the operation.		

Activities

Actor		**System**	
Main flow: Deleted successfully			
1	Administrator clicks the "Delete" button		
		2	Displays a dialog box to confirm the action
3	Administrator clicks the "Yes" button on the dialog box		
		4	Delete the version of the account with that ID in the database

Business Rules

Rule no.	**Rule**	**Description**

System message

Message no.	Message	Case
1	"Are you sure you want to delete this account? ”	After the administrator clicks on the "Delete" button
2	"Deleted successfully!"	After completing deleting that account from the database

3.6.UC_TRNEE_01: VIEW PERSONAL COURSES:

Use Case Description

Name	View personal courses list	**ID**	UC_TREE_01
Description	Allow the trainee to view his/her personal courses list		
Actor	Trainee	**Trigger**	Trainee logs in to the system
Pre-condition	The user logged in with correct trainee account		
Post condition			

Activities

Actor		System	
Main flow: Display course list successfully			
1	Trainee logged in successfully to the system		
		2	Move to the “Course Index” page

Business Rules

Rule no.	Rule	Description
1	Trainee can view only their courses, course details, topics in that course, and the name of the instructor for each topic.	Trainee can only view their courses, course details, topics in that course and the name of the teacher of each topic

System message

Message no.	Message	Case

3.7.UC_TRNER_01: VIEW PERSONAL COURSES

Use Case Description

Name	View personal courses list	**ID**	UC_TRER_01
Description	Allow the trainer to view his/her personal courses list		
Actor	Trainer	**Trigger**	Trainer log in to the system

			The trainer clicks on the "Overview page" button
Pre-condition	User logged in with the correct trainer account		
Post condition			

Activities

Actor		**System**	
Main flow: Display course list successfully			
1	The Trainer has logged in successfully. Or Coach has clicked the button "Dashboard"		
		2	Move to the "Course Index" page

Business Rules

Rule no.	**Rule**	**Description**
1	Trainer can view only their courses, course details, topics in that course, and the name of the instructor for each topic.	Trainer can view only their courses, course details, topics in that course, and the name of the instructor for each topic.

System message

Message no.	**Message**	**Case**

3.8.UC_TRNER_02: UPDATE TRAINER PROFILE:

Use Case Description

Name	Login	**ID**	UC_TRER_02
Description	Allow trainer to modify their personal information		
Actor	Trainer	**Trigger**	User clicks "Update profile" button
Pre-condition	The user logged in with correct trainer account		
Post condition			

Activities

Actor		**System**	
Main flow: Display course list successfully			
1	User clicks on "Modify profile" button		
		2	Redirect to "Edit" page
3	The user modifies his / her personal information		
4	User clicks on "Save		
		5	Update instance with the same trainer ID in the database using the inserted trainer data

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case
1	“The Trainer name field cannot be left blank!”	Trainer forgot to refill their name
2	"Successfully edited!"	Successfully updated to database

3.9.UC_STF_CO_01: VIEW COURSE LIST

Use Case Description

Name	View courses list	**ID**	UC_STF_C01
Description	Allow the training staff to view courses list		
Actor	Staff	**Trigger**	Training staff logs in to the system Trainer click “Course” button
Pre-condition	The user logged in with correct training staff account		
Post condition			

Activities

Actor		System	
Main flow: Display course list successfully			
1	The staff successfully logged in to the system Or The staff click on the button "Course"		
		2	Move to the “Course Index” page

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case

3.10. UC_STAFF_CO_02: CREATE NEW COURSE:

Use Case Description

Name	Create new course	**ID**	UC_STF_C02
Description	Allow Training staff to create course		
Actor	Training staff	**Trigger**	Training staff clicks the “Create new” button on “Course Index” page
Pre-condition	The Course Index page loads completely		
Post condition			

Activities

Actor		**System**	
Main flow: Display course list successfully			
1	Training Staff Click the "Create New" button		
		2	Move to the "Create new" page
3	The training staff inserts information about the new course he/she wants to create		
4	The training staff click the "Create		
		5	Create a new course version in the database using the inserted data
		6	Print success message

Business Rules

Rule no.	**Rule**	**Description**
1	The course name is the only version	Every course needs to be a unique chemist.

System message

Message no.	**Message**	**Case**
1	"The course name field cannot be left blank!"	Training staff forgot to enter the course name
		Successfully updated database

3.11. UC_STAFF_CO_03: EDIT EXISTING COURSE:

Use Case Description

Name	Edit existing course	**ID**	UC_STF_C03
Description	Allow user to edit existing course		
Actor	Staff	**Trigger**	The user clicks the “Edit” button in “Course Index” page
Pre-condition	The Course Index page loads completely		
Post condition	Edit existing course		

Activities

Actor	**System**

Main flow: Display course list successfully			
1	Staff click the button "edit "		
		2	Move to page "Edit"
3	Staff inserts new information of the existing course that they wish to modify		
4	Staff Training Click the button "save		
		5	Update course version in the database using inserted data
		6	Print success message

Business Rules

Rule no.	Rule	Description
1	Name Edit can the same old Name	Course names have exist.

System message

Message no.	Message	Case
1	"The course name field cannot be left blank!"	Training staff forgot to enter the course name
		Successfully updated database

3.12. UC_STAFF_CO_04: DELETE EXISTING COURSE

Use Case Description

Name	Delete existing course	**ID**	UC_STF_C04
Description	Allow user to delete an existing course		
Actor	Training staff	**Trigger**	User clicks the "Delete" button on Course Index page
Pre-condition	The Course Index page loads completely		
Post condition	Delete existing course		

Activities

Actor		**System**	
Main flow: Display course list successfully			
1	Staff click the button "Delete "		
		2	Display confirm the action
3	Staff Click button Yes		
		4	Delete course for that Id in the Database

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case
1	"Are you sure you want to delete this Course? "	After staff click button Delete
		After completing deleting that Course from the database

3.13. UC_STAFF_CO_06: ASSIGN TOPIC TO COURSES

Use Case Description

Name	Assign topics to course	**ID**	UC_STF_C05
Description	Allow user to assign topics to course		
Actor	Staff	**Trigger**	User clicks the "Add topics" button from "Course Index" page
Pre-condition	The Course Index page loads completely		
Post condition	Assign topics to course		

Activities

Actor		**System**	
Main flow: Add topics successfully			
1	The user clicks button "Add topic..." From the "Course Index" page		
		2	Redirect user to "Add topics" page
		3	Display all available topics as checkboxes
4	Users tick the topics they want to add to the course		
5	User clicks the "Save		
		6	Create new instances of "Course Details" in the database
		7	Display success message

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case

3.14. UC_STF_CO_01: VIEW COURSE CATEGORY LIST

Use Case Description

Name	View courses list	**ID**	UC_STF_CC01
Description	Allow the training staff to view courses list		
Actor	Staff	**Trigger**	Training staff logs in to the system Trainer click "Course Category" button
Pre-condition	The user logged in with correct staff account		
Post condition			

Activities

Actor		**System**	
Main flow: Display course list successfully			
1	The staff successfully logged in to the system Or The staff click on the button "Course Category"		
		2	Move to the "Course Category Index" page

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case

3.15. UC_STAFF_CO_02: CREATE NEW COURSE CATEGORY:

Use Case Description

Name	Create new course category	**ID**	UC_STF_CC02
Description	Allow staff to create course category		
Actor	Staff	**Trigger**	Training staff clicks the "Create new" button on "Course Category Index" page
Pre-condition	The Course Category Index page loads completely		

Post condition	

Activities

Actor		**System**	
Main flow: Display course list successfully			
1	Training Staff Click the "Create New" button		
		2	Move to the "Create new" page
3	The training staff inserts information about the new course category they wants to create		
4	The training staff click the "Create"		
		5	Create a new course category version in the database using the inserted data
		6	Print success message

Business Rules

Rule no.	**Rule**	**Description**
1	The course category name is the only version	Every course category name needs to be a unique chemist.

System message

Message no.	**Message**	**Case**
1	"The course category name field cannot be left blank!"	Staff forgot to enter the course category name
		Successfully updated database

3.16. UC_STAFF_CO_03: EDIT EXISTING COURSE CATEGORY:

Use Case Description

Name	Edit existing course category	**ID**	UC_STF_CC03
Description	Allow user to edit existing course category		
Actor	Staff	**Trigger**	The user clicks the "Edit" button in "Course category Index" page
Pre-condition	The Course Category Index page loads completely		
Post condition	Edit existing course category		

Activities

Actor		**System**	
Main flow: Display course category list successfully			
1	Staff click the button "edit "		
		2	Move to page "Edit"

3	Staff inserts new information of the existing course category that they wish to modify		
4	Staff Click the button "save		
		5	Update course version in the database using inserted data
		6	Print success message

Business Rules

Rule no.	**Rule**	**Description**
1	Name Edit can the same old Name	Course category names have exist.

System message

Message no.	**Message**	**Case**
1	"The course category name field cannot be left blank!"	Staff forgot to enter the course category name
		Successfully updated database

3.17. UC_STAFF_CO_04: DELETE EXISTING COURSE CATEGORY

Use Case Description

Name	Delete existing course category	**ID**	UC_STF_CC04
Description	Allow user to delete an existing course category		
Actor	Staff	**Trigger**	User clicks the “Delete” button on Course category Index page
Pre-condition	The Course category Index page loads completely		
Post condition	Delete existing course		

Activities

Actor		**System**	
Main flow: Display course category list successfully			
1	Staff click the button "Delete "		
		2	Display confirm the action
3	Staff Click button Yes		
		4	Delete course category for that Id in the Database

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case
1	"Are you sure you want to delete this Course category? ”	After staff click button Delete
		After completing deleting that Course Category from the database

3.18. UC_STF_CO_01: VIEW TOPIC LIST

Use Case Description

Name	View topics list	**ID**	UC_STF_T01
Description	Allow the staff to view topic list		
Actor	Staff	**Trigger**	staff logs in to the system Trainer click “Topic” button
Pre-condition	The user logged in with correct staff account		
Post condition			

Activities

Actor		System	
Main flow: Display course list successfully			
1	The user insert the Username and Password to proper textbox and click the login button		
		2	Check the validation of Username and Password
		3	Check the role true
		4	Move to the default page based on User role

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case
1	Wrong account please check again!	Wrong Username/ Password

3.19. UC_STAFF_CO_02: CREATE NEW TOPIC

Use Case Description

Name	Create new topic	**ID**	UC_STF_T02
Description	Allow staff to create topic		
Actor	Staff	**Trigger**	Training staff clicks button "Create new" on the topic index
Pre-condition	The Topic Index page loads completely		
Post condition			

Activities

Actor		**System**	
Main flow: Create Topic Done			
1	Staff Click the "Create New" button		
		2	Move to the "Create new" page
3	Staff inserts information about the new topic they want		
4	Staff click the "Create"		
		5	Create a new topic version in the database using the inserted data
		6	Print success message

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case
1	"The topic name field cannot be left blank!"	Training staff forgot to enter the topic name
		Successfully updated database

3.20. UC_STAFF_CO_03: EDIT EXISTING TOPIC:

Use Case Description

Name	Edit Topic	**ID**	UC_STF_T03
Description	Allow Staff to edit Topic		

Actor	Staff	**Trigger**	The staff clicks Button Edit in the Topic Index
Pre-condition	The Topic Index page loads completely		
Post condition	Edit Topic		

Activities

Actor		**System**	
Main flow: Edit Topic Done!			
1	Staff click the button "edit "		
		2	Move to page “Edit”
3	Staff inserts new information of the existing topic that they wish to modify		
4	Staff Training Click the button "save		
		5	Update topic version in the database using inserted data
		6	Print success message

Business Rules

Rule no.	**Rule**	**Description**

System message

Message no.	**Message**	**Case**
1	"The Topic name field cannot be left blank!"	Training staff forgot to enter the topic name
		Successfully updated database

3.21. UC_STAFF_CO_04: DELETE EXISTING TOPIC

Use Case Description

Name	Delete Topic	**ID**	UC_STF_T04
Description	Allow user to delete an Topic		
Actor	Staff	**Trigger**	User clicks the Button Delete on the Topic Index
Pre-condition	The Topic Index page loads completely		
Post condition	Delete Topic		

Activities

Actor	**System**
Main flow: Display topic list successfully	

1	Staff click the button "Delete "		
		2	Display confirm the action
3	Staff Click button Yes		
		4	Delete topic for that Id in the Database

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case
1	"Are you sure you want to delete this Topic? ”	After staff click button Delete
		After completing deleting that Topic from the database

3.22. UC_STF_CO_01: VIEW TRAINER

Use Case Description

Name	View trainer list	**ID**	UC_STF_TR01
Description	Allow the staff to view trainer list		
Actor	Staff	**Trigger**	staff logs in to the system Trainer click “Topic” button
Pre-condition	The user logged in with correct account		
Post condition			

Activities

Actor		System	
Main flow: Display Trainer list successfully			
1	The user insert the Username and Password to proper textbox and click the login button		
		2	Check the validation of Username and Password
		3	Check the role true
		4	Move to the default page based on User role

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case
1	Wrong account please check again!	Wrong Username/ Password

3.23. UC_STAFF_CO_02: CREATE NEW TRAINER

Use Case Description

Name	Create new trainer	**ID**	UC_STF_TR02
Description	Allow staff to create Trainer		
Actor	Staff	**Trigger**	Training staff clicks button “Create new” on the Trainer index
Pre-condition	The Trainer Index page loads completely		
Post condition			

Activities

Actor		**System**	
Main flow: Create Trainer Done			
1	Staff Click the "Create New" button		
		2	Move to the "Create new" page
3	Staff inserts information about the new topic they want		
4	Staff click the "Create”		
		5	Create a new Trainer version in the database using the inserted data
		6	Print success message

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case
1	"The Trainer name field cannot be left blank!"	Staff forgot to enter the course name
		Successfully updated database

3.24. UC_STAFF_CO_03: EDIT EXISTING TRAINER:

Use Case Description

Name	Edit Trainer	**ID**	UC_STF_TR03
Description	Allow Staff to edit Trainer		
Actor	Staff	**Trigger**	The staff clicks Button Edit in the Trainer Index
Pre-condition	The Trainer Index page loads completely		
Post condition	Edit Trainer		

Activities

Actor		**System**	
Main flow: Edit Trainer Done!			
1	Staff click the button "edit "		
		2	Move to page “Edit”
3	Staff inserts new information of the existing Trainer that they wish to modify		
4	Staff Training Click the button "save		
		5	Update Trainer version in the database using inserted data
		6	Print success message

Business Rules

Rule no.	**Rule**	**Description**

System message

Message no.	**Message**	**Case**
1	"The Trainer name field cannot be left blank!"	Staff forgot to enter the course name
		Successfully updated database

3.25. UC_STAFF_CO_04: DELETE EXISTING TRAINER

Use Case Description

Name	Delete Trainer	**ID**	UC_STF_TR04
Description	Allow user to delete an Trainer		
Actor	Staff	**Trigger**	User clicks the Button Delete on the Trainer Index
Pre-condition	The Trainer Index page loads completely		

Post condition	Delete Trainer

Activities

Actor		**System**	
Main flow: Display Trainer successfully			
1	Staff click the button "Delete "		
		2	Display confirm the action
3	Staff Click button Yes		
		4	Delete Trainer for that Id in the Database

Business Rules

Rule no.	**Rule**	**Description**

System message

Message no.	**Message**	**Case**
1	"Are you sure you want to delete this Trainer? ”	After staff click button Delete
		After completing deleting that Trainer from the database

3.26. UC_STF_CO_01: VIEW TRAINEE

Use Case Description

Name	View trainee list	**ID**	UC_STF_TE01
Description	Allow the staff to view trainee list		
Actor	Staff	**Trigger**	staff logs in to the system Trainee click “Topic” button
Pre-condition	The user logged in with correct account		
Post condition			

Activities

Actor		**System**	
Main flow: Display Trainee list successfully			
1	The user insert the Username and Password to proper textbox and click the login button		
		2	Check the validation of Username and Password
		3	Check the role true
		4	Move to the default page based on User role

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case
1	Wrong account please check again!	Wrong Username/ Password

3.27. UC_STAFF_CO_02: CREATE NEW TRAINEE

Use Case Description

Name	Create new trainee	**ID**	UC_STF_TE02
Description	Allow staff to create Trainee		
Actor	Staff	**Trigger**	Training staff clicks button Create new on the Trainee Page
Pre-condition	The Trainee Index page loads completely		
Post condition			

Activities

Actor		**System**	
Main flow: Create Trainee Done			
1	Staff Click the "Create New" button		
		2	Move to the "Create new" page
3	Staff inserts information about the new Trainee they want		
4	Staff click the "Create"		
		5	Create a new Trainee version in the database using the inserted data
		6	Print success message

Business Rules

Rule no.	Rule	Description

System message

Message no.	Message	Case
1	"The Trainee name field cannot be left blank!"	Staff forgot to enter the trainee name
		Successfully updated database

3.28. UC_STAFF_CO_03: EDIT EXISTING TRAINEE:

Use Case Description

Name	Edit Trainee	**ID**	UC_STF_TE03
Description	Allow Staff to edit Trainee		
Actor	Staff	**Trigger**	The staff clicks Button Edit in the Trainee Index
Pre-condition	The Trainee Index page loads completely		
Post condition	Edit Trainee		

Activities

Actor		**System**	
Main flow: Edit Trainee Done!			
1	Staff click the button "edit "		
		2	Move to page “Edit”
3	Staff inserts new information of the existing Trainee that they wish to modify		
4	Staff Training Click the button "save		
		5	Update Trainee version in the database using inserted data
		6	Print success message

Business Rules

Rule no.	**Rule**	**Description**

System message

Message no.	**Message**	**Case**
1	"The Trainee name field cannot be left blank!"	Staff forgot to enter the course name
		Successfully updated database

3.29. UC_STAFF_CO_04: DELETE EXISTING TRAINEE

Use Case Description

Name	Delete Trainee	**ID**	UC_STF_TR04
Description	Allow user to delete an Trainee		

Actor	Staff	**Trigger**	User clicks the Button Delete on the Trainee Index
Pre-condition	The Trainee Index page loads completely		
Post condition	Delete Trainee		

Activities

Actor		**System**	
Main flow: Display Trainee successfully			
1	Staff click the button "Delete "		
		2	Display confirm the action
3	Staff Click button Yes		
		4	Delete Trainee for that Id in the Database

Business Rules

Rule no.	**Rule**	**Description**

System message

Message no.	**Message**	**Case**
1	"Are you sure you want to delete this Trainee? ”	After staff click button Delete
		After completing deleting that Trainer from the database

3.30. UC_STAFF_CO_06: ASSIGN TRAINEE TO COURSE

Use Case Description

Name	Assign Trainee to course	**ID**	UC_STF_TE05
Description	Allow user to assign trainee to course		
Actor	Staff	**Trigger**	User clicks the “Add trainee” button from “Course Index” page
Pre-condition	The Course Index page loads completely		
Post condition	Assign trainee to course		

Activities

Actor		**System**	
Main flow: Add topics successfully			
1	The user clicks button "Add trainee..." From the "Course Index" page		

		2	Redirect user to "Add trainee" page
		3	Display all available Trainee as checkboxes
4	Users tick the topics they want to add to the course		
5	User clicks the "Save		
		6	Create new instances of "Enrollment" in the database
		7	Display success message

Business Rules

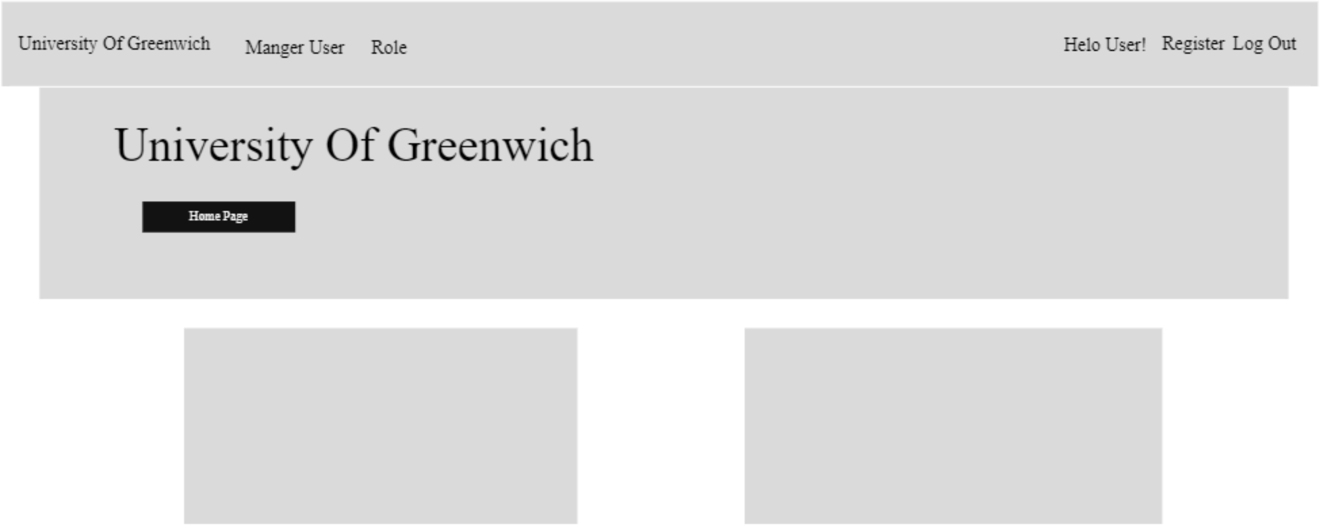
Rule no.	Rule	Description

System message

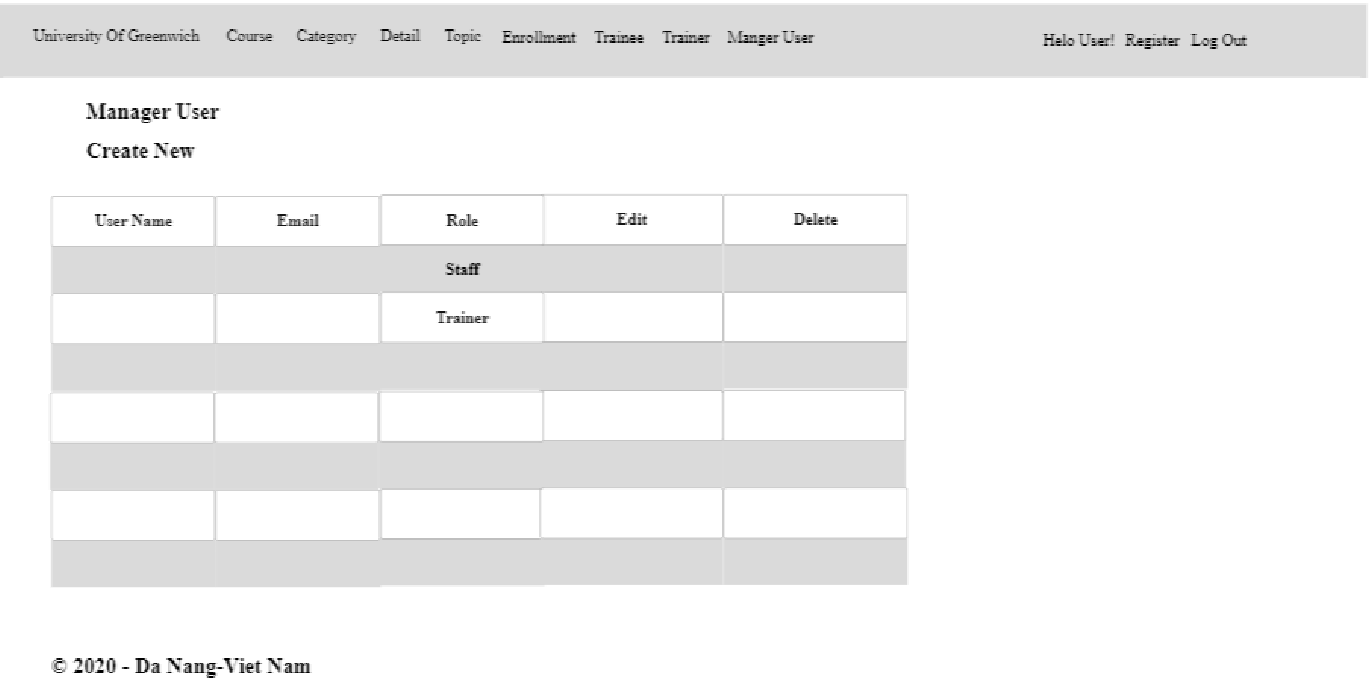
Message no.	Message	Case

4. MOCKUP WIREFRAMES

Home Page



Manager User & Role



University Of Greenwich Course Category Detail Topic Enrollment Trainee Trainer Manger User Helo User! Register Log Out

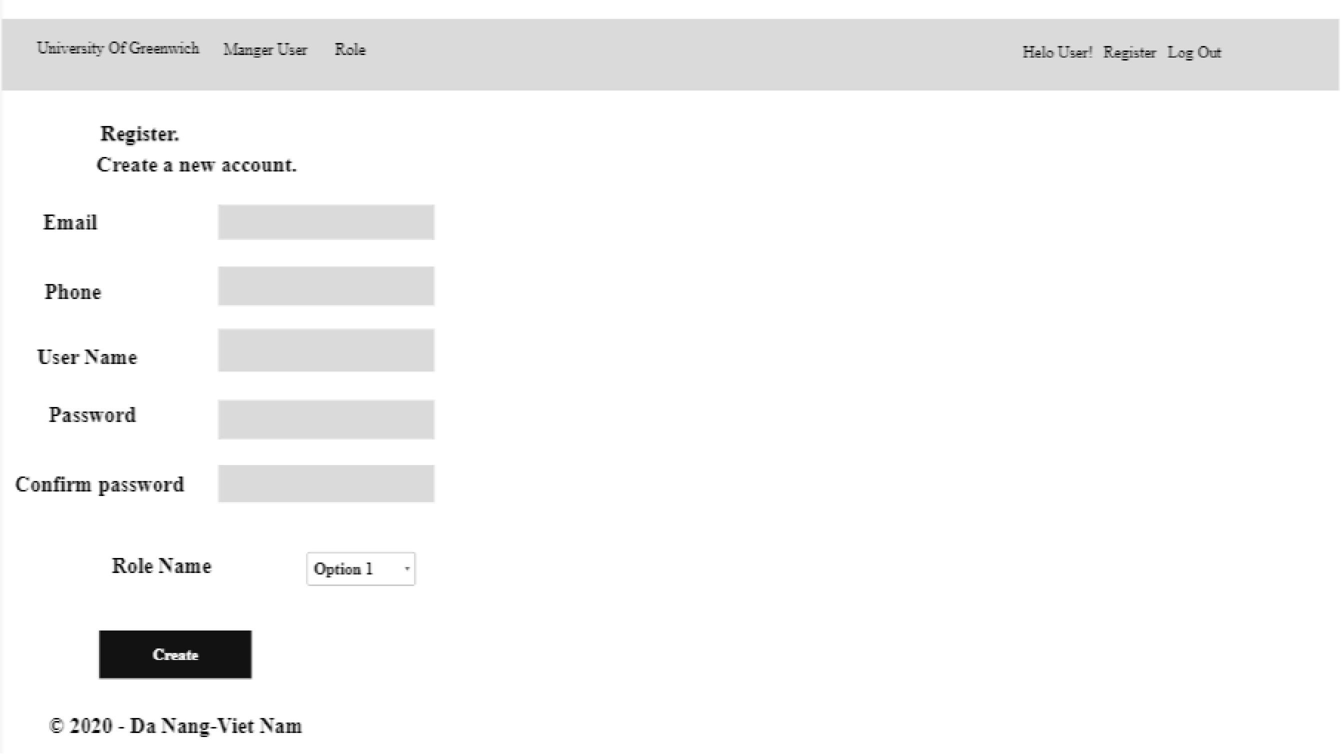
Manager User

Create New

User Name	Email	Role	Edit	Delete
		Staff		
		Trainer		

© 2020 - Da Nang-Viet Nam

Register User



University Of Greenwich Manger User Role Helo User! Register Log Out

Register.
Create a new account.

Email

Phone

User Name

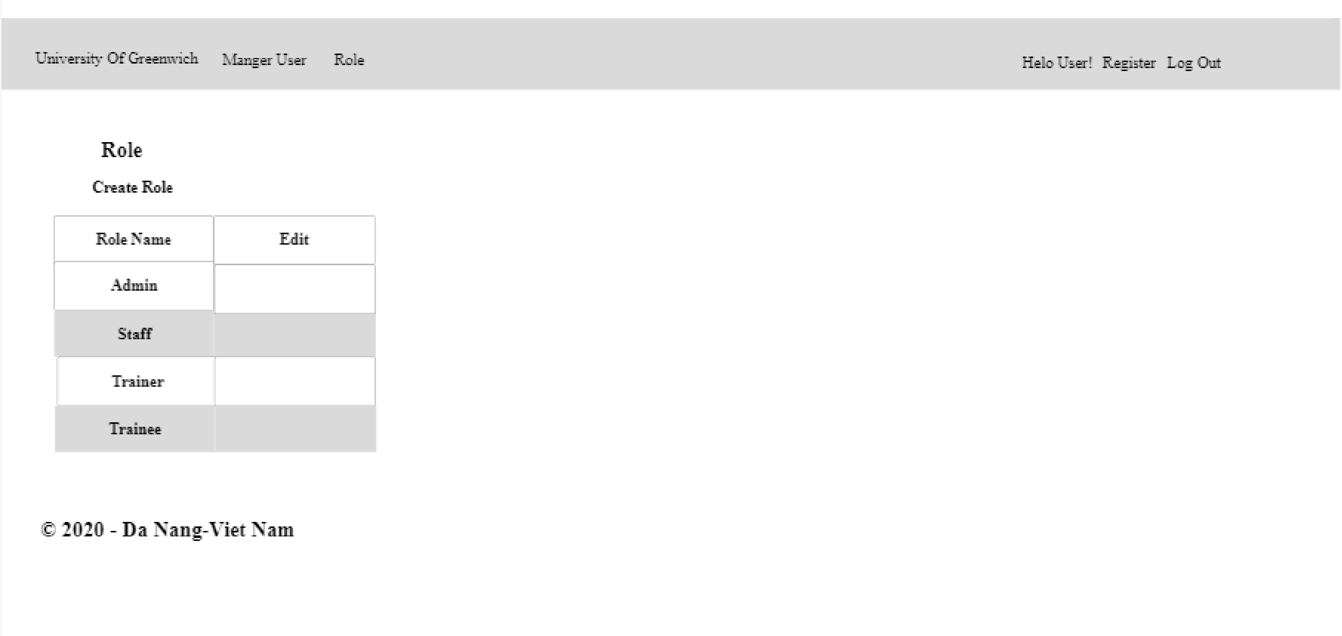
Password

Confirm password

Role Name Option 1

© 2020 - Da Nang-Viet Nam

Role Index



University Of Greenwich Manger User Role

Helo User! Register Log Out

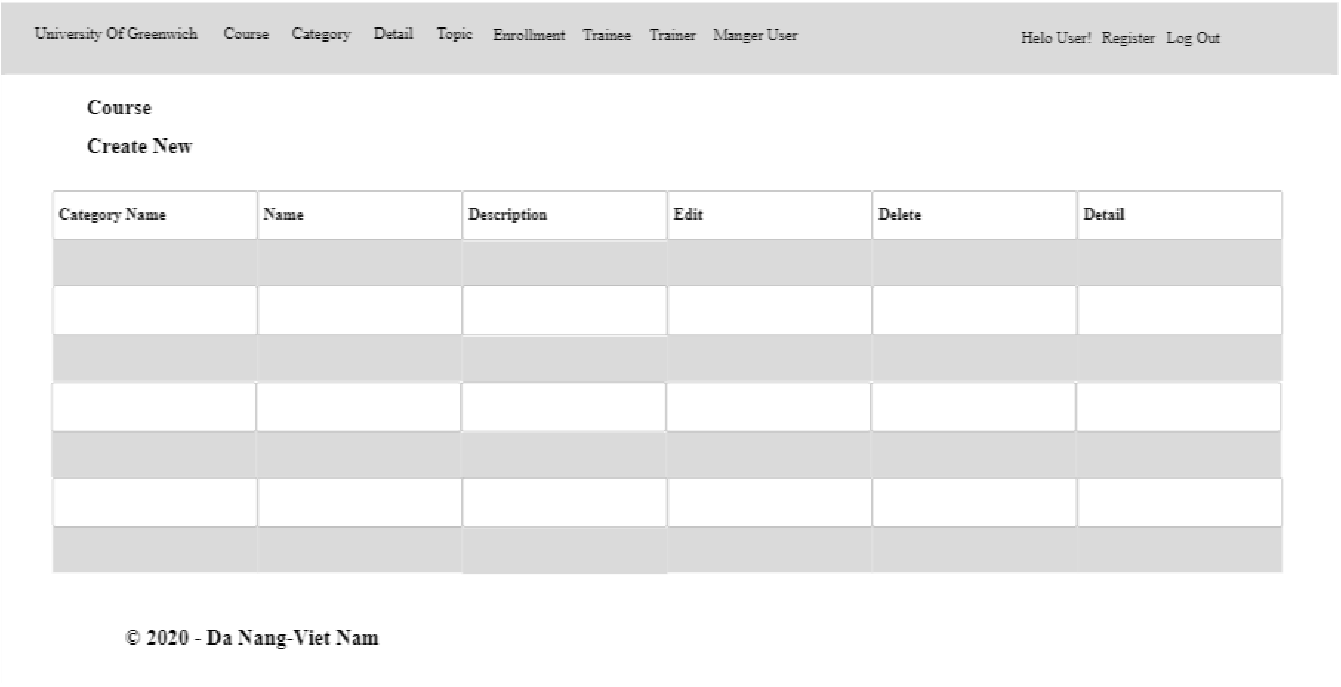
Role

Create Role

Role Name	Edit
Admin	
Staff	
Trainer	
Trainee	

© 2020 - Da Nang-Viet Nam

Course Index



University Of Greenwich Course Category Detail Topic Enrollment Trainee Trainer Manger User

Helo User! Register Log Out

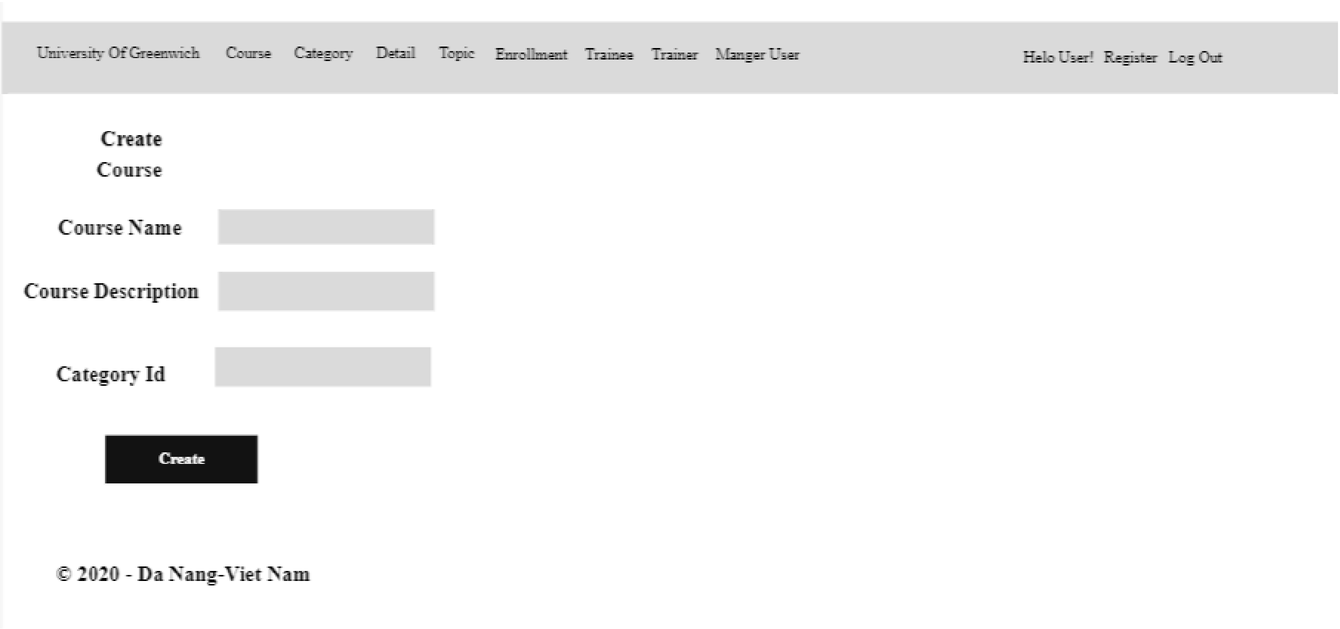
Course

Create New

Category Name	Name	Description	Edit	Delete	Detail

© 2020 - Da Nang-Viet Nam

Create Course



Create Course

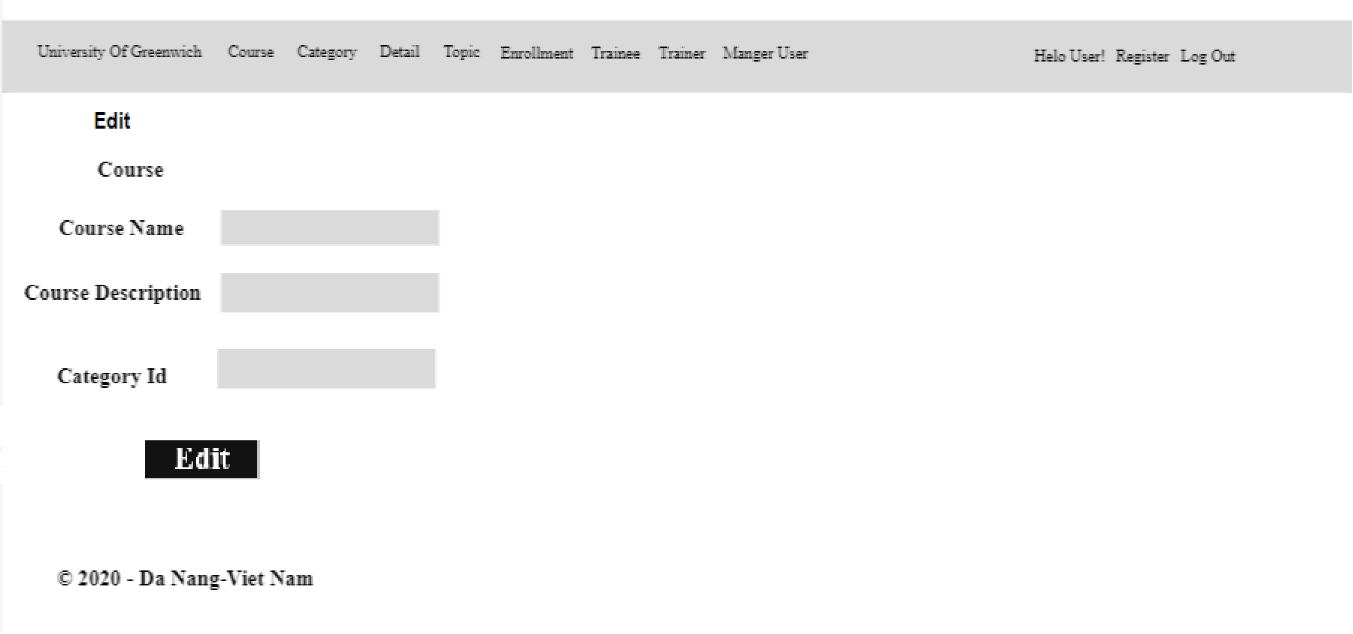
Course Name

Course Description

Category Id

© 2020 - Da Nang-Viet Nam

Edit course



Edit

Course

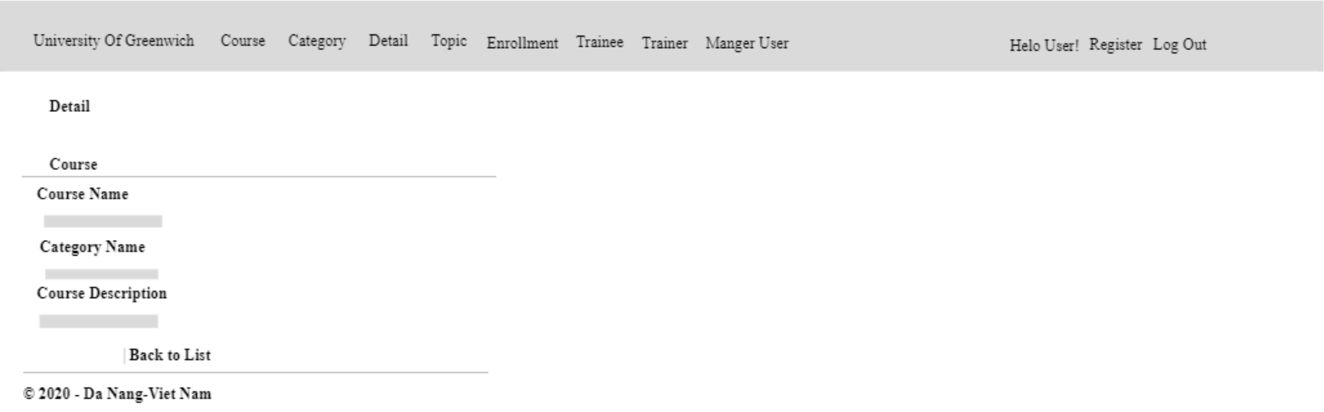
Course Name

Course Description

Category Id

© 2020 - Da Nang-Viet Nam

View Detail Course



Detail

Course

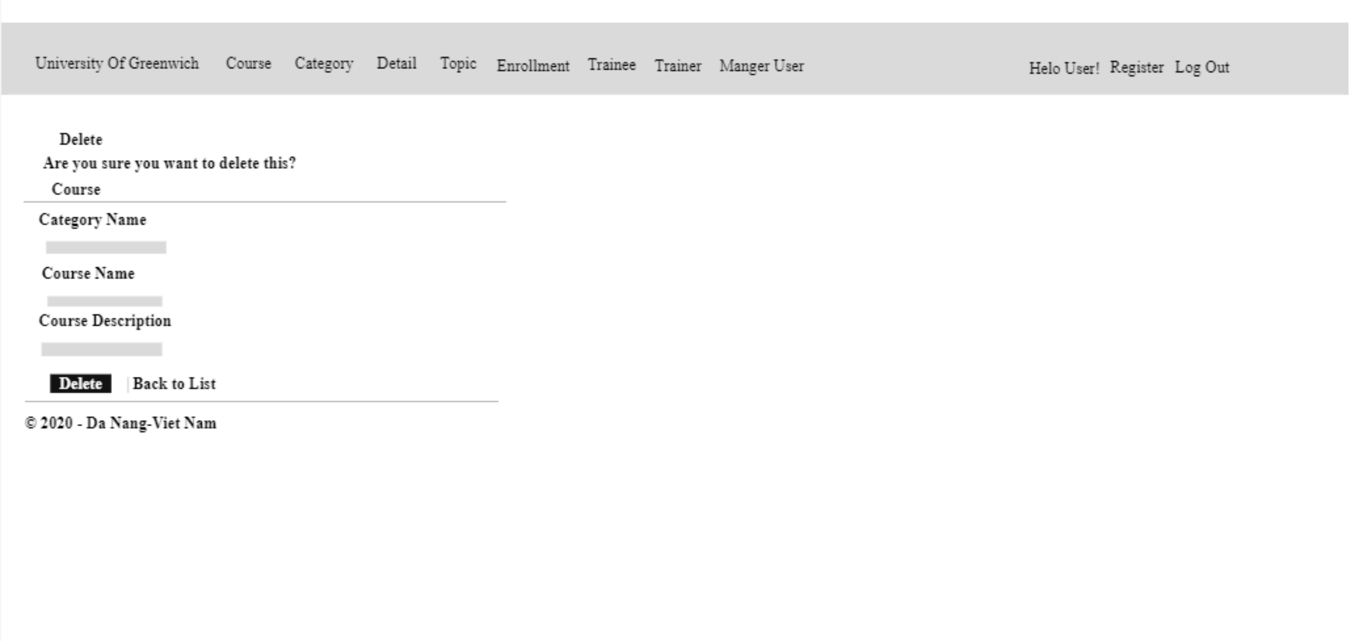
Course Name

Category Name

Course Description

© 2020 - Da Nang-Viet Nam

Delete Course



Delete

Are you sure you want to delete this?

Course

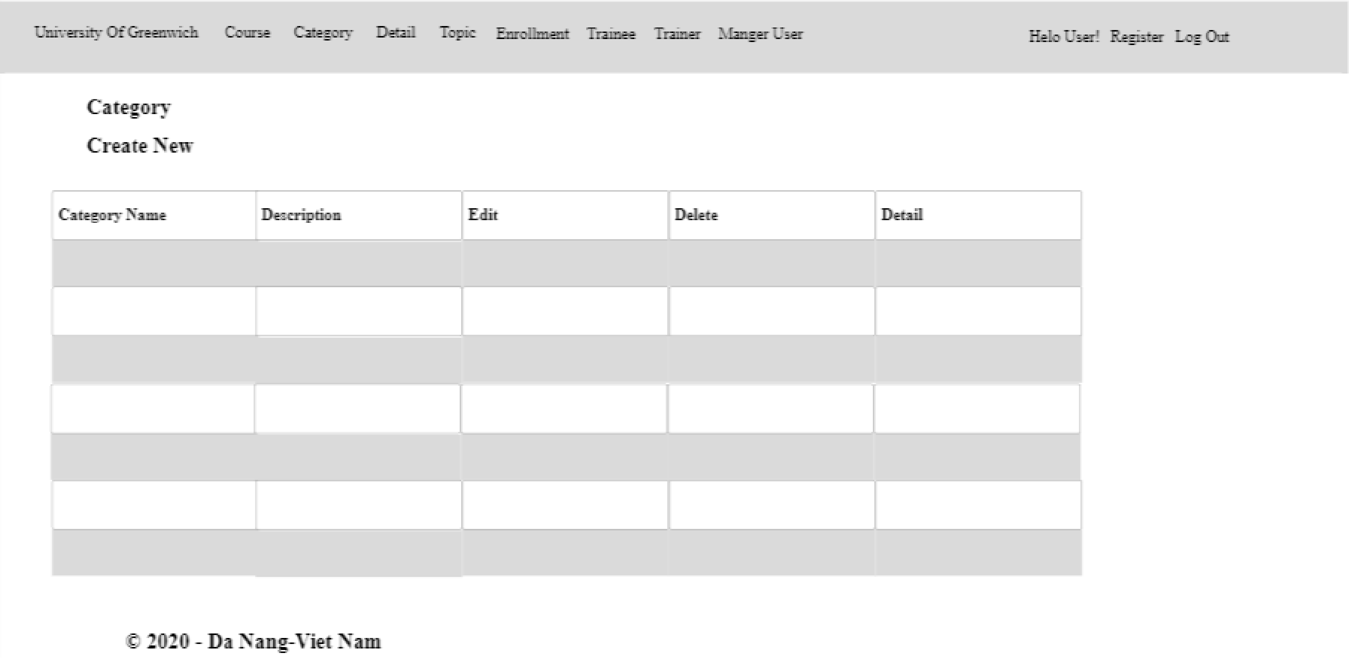
Category Name

Course Name

Course Description

© 2020 - Da Nang-Viet Nam

Category Index



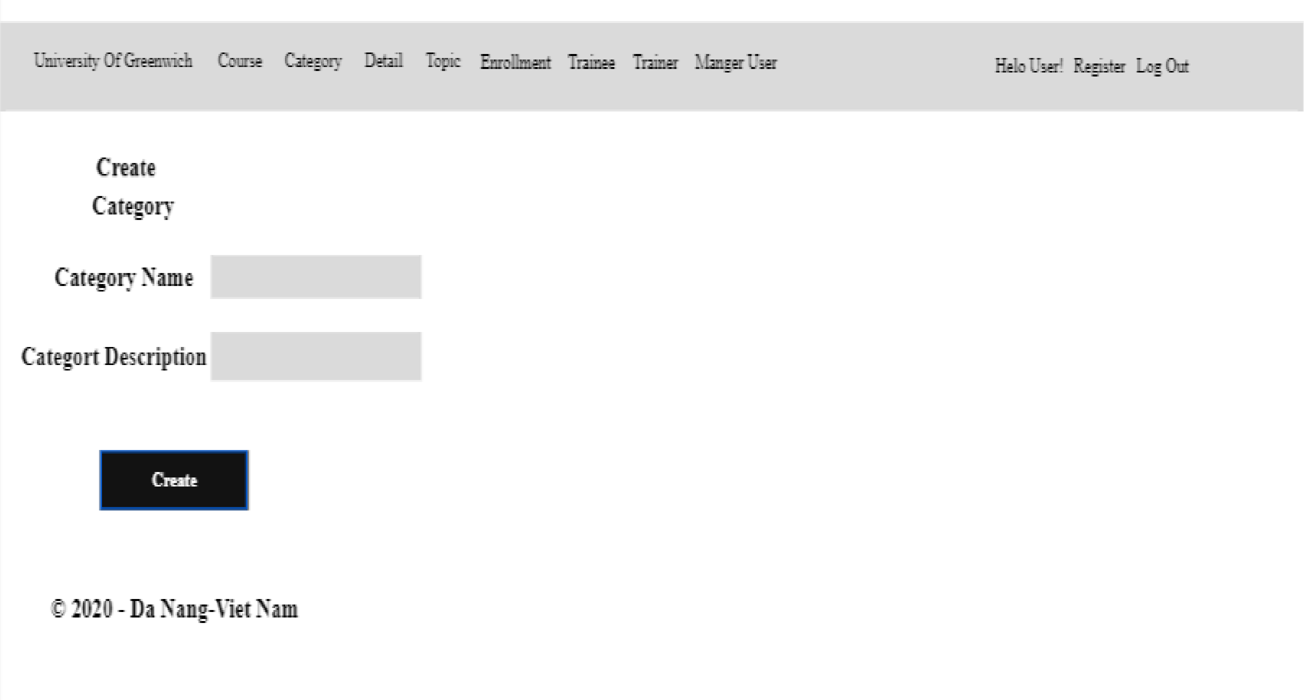
Category

Create New

Category Name	Description	Edit	Delete	Detail

© 2020 - Da Nang-Viet Nam

Create Category Page



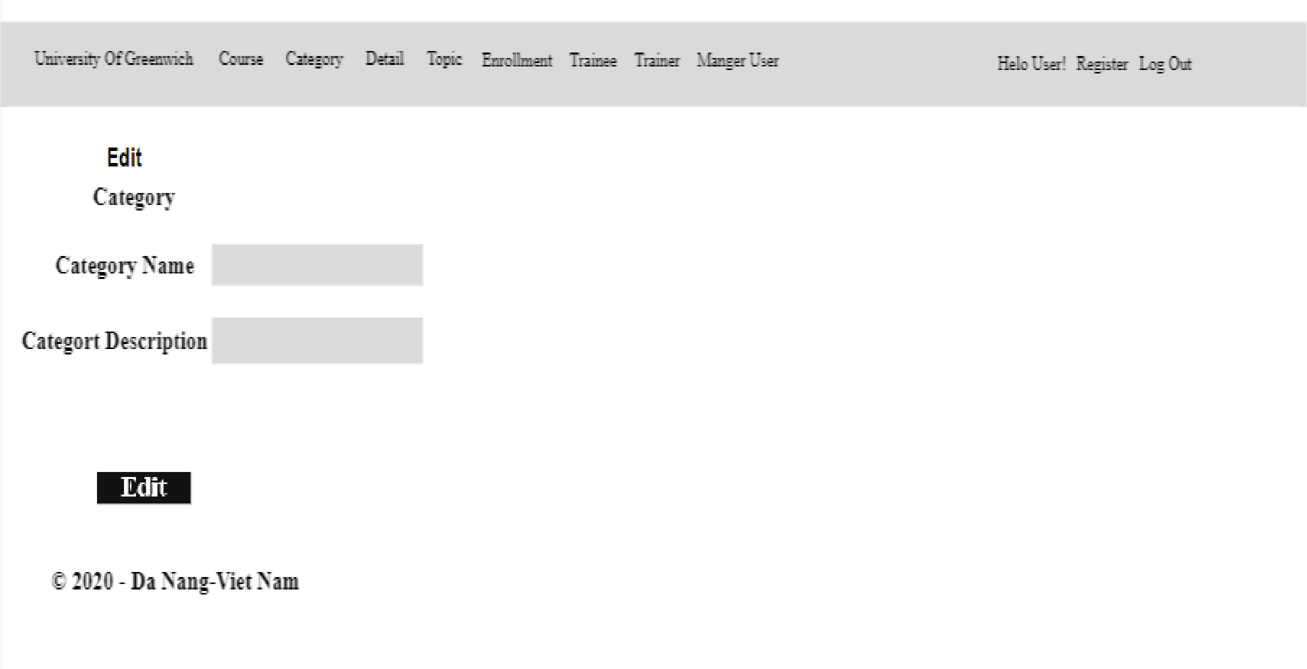
Create
Category

Category Name

Categort Description

© 2020 - Da Nang-Viet Nam

Edit Category Page



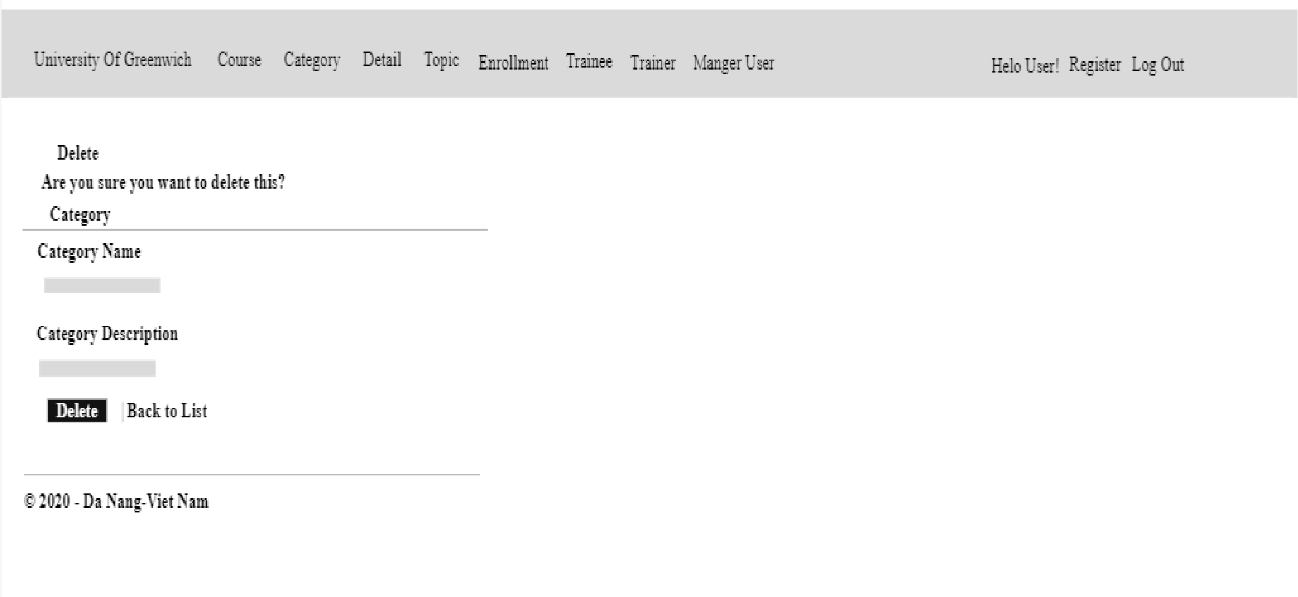
Edit
Category

Category Name

Categort Description

© 2020 - Da Nang-Viet Nam

Delete Category Page



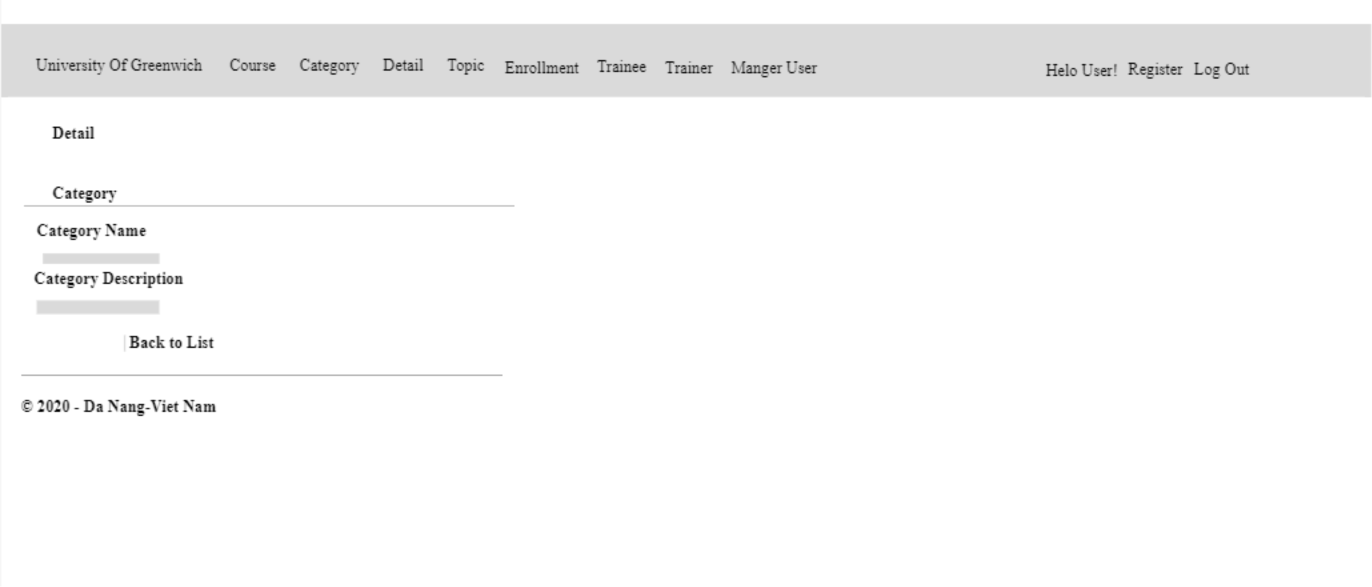
Delete
Are you sure you want to delete this?
Category

Category Name

Category Description

© 2020 - Da Nang-Viet Nam

Detail Category Page



University Of Greenwich Course Category Detail Topic Enrollment Trainee Trainer Manger User Helo User! Register Log Out

Detail

Category

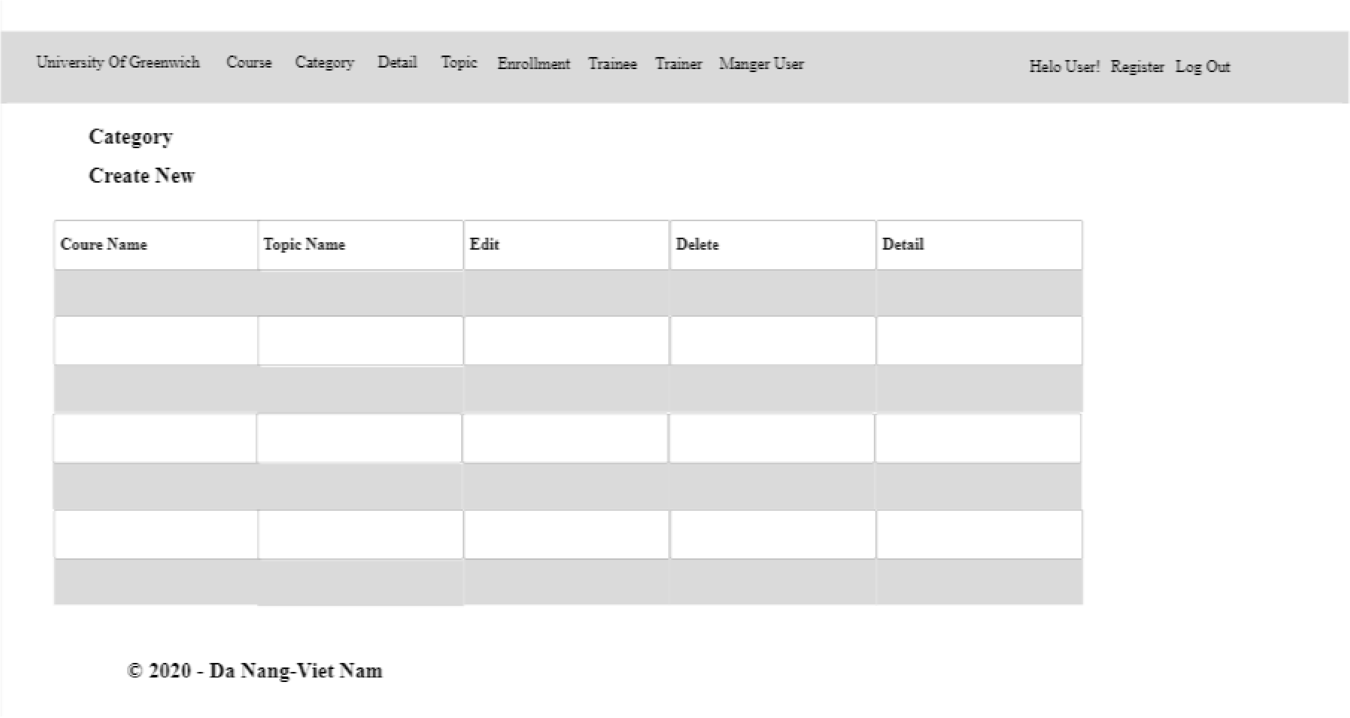
Category Name

Category Description

Back to List

© 2020 - Da Nang-Viet Nam

Detail Index Page



University Of Greenwich Course Category Detail Topic Enrollment Trainee Trainer Manger User Helo User! Register Log Out

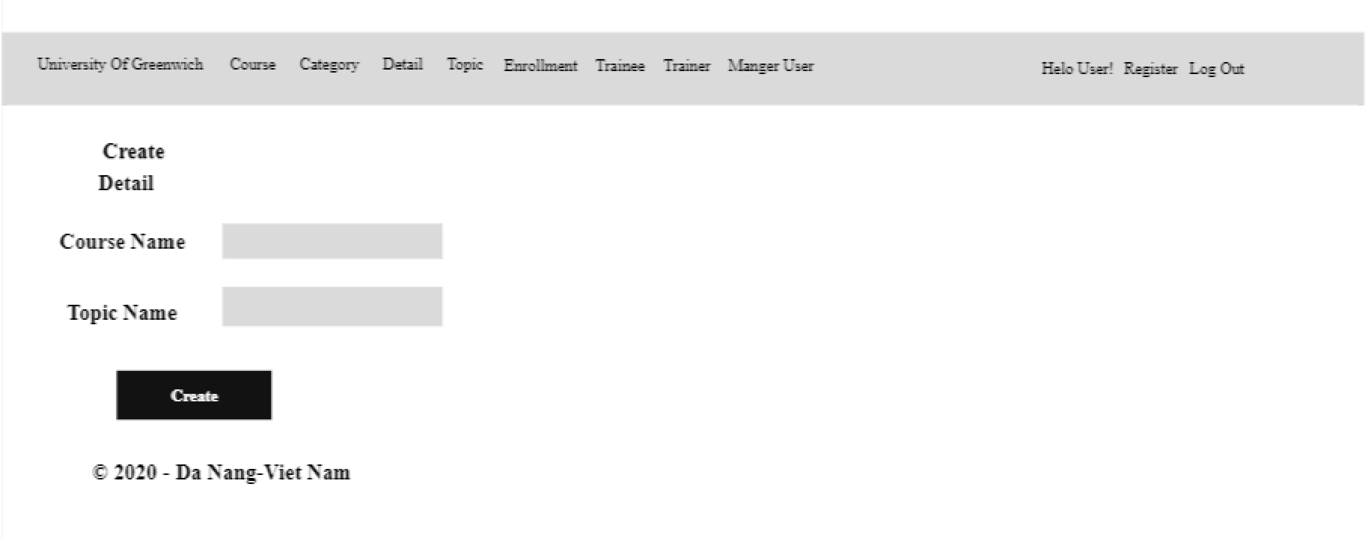
Category

Create New

Coure Name	Topic Name	Edit	Delete	Detail

© 2020 - Da Nang-Viet Nam

Create Detail Page



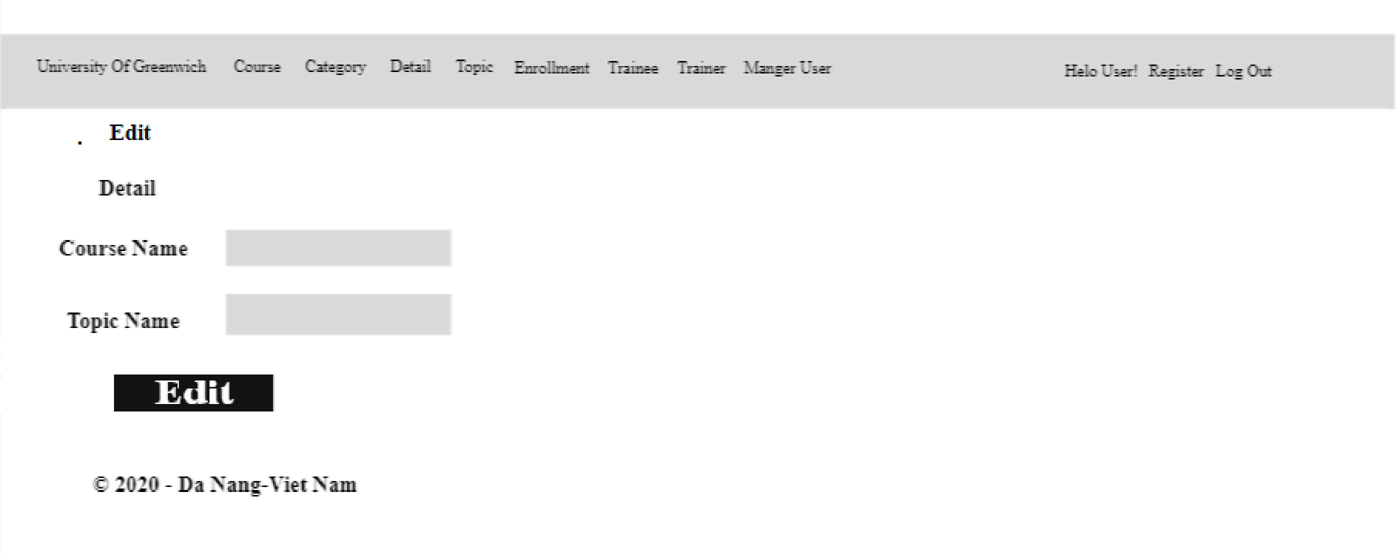
Create Detail

Course Name

Topic Name

© 2020 - Da Nang-Viet Nam

Edit Detail Page



Edit

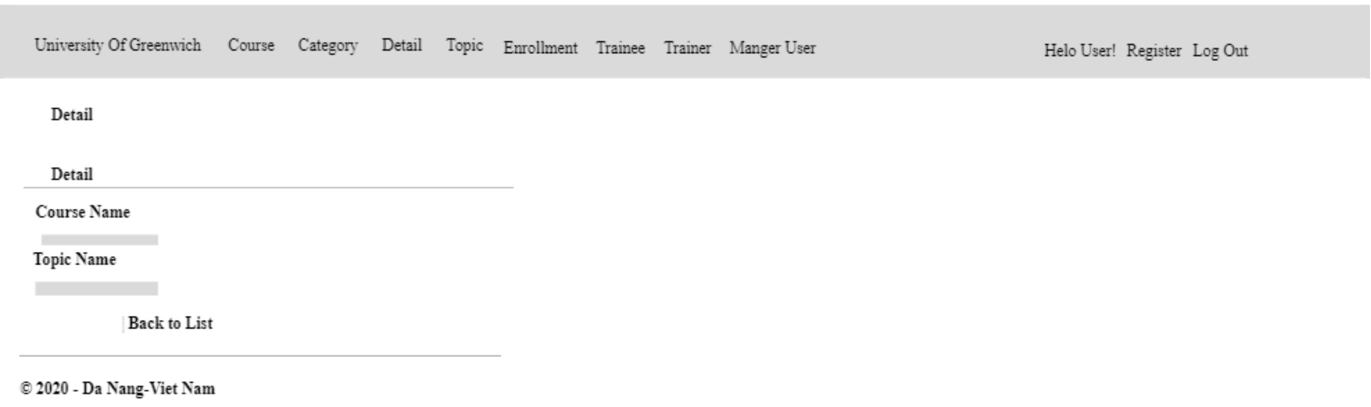
Detail

Course Name

Topic Name

© 2020 - Da Nang-Viet Nam

Detail for Detail Page



Detail

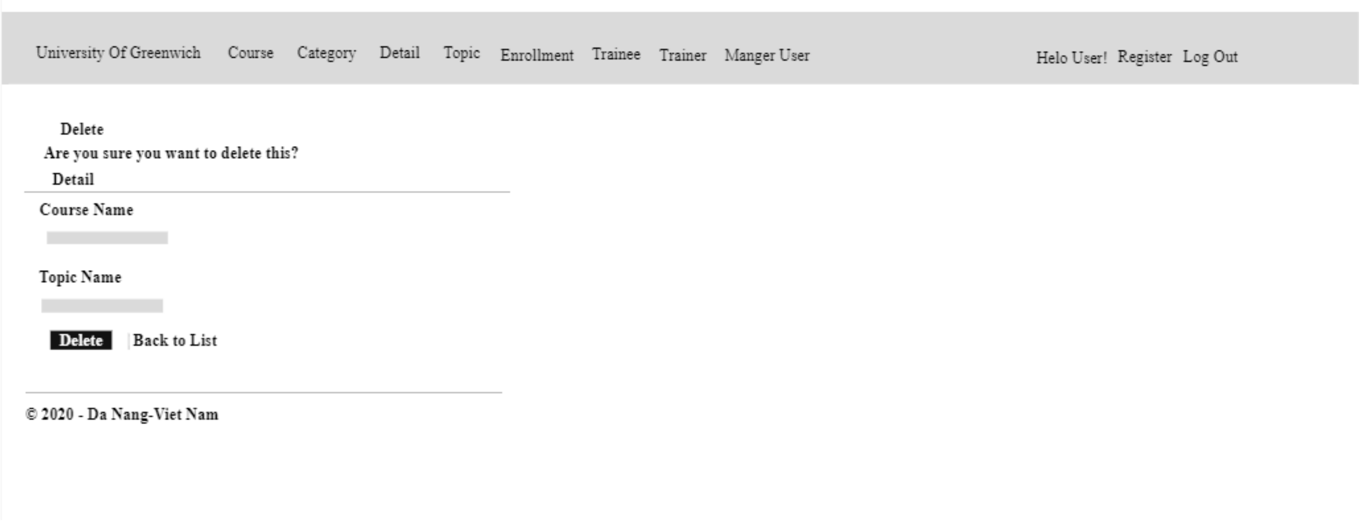
Detail

Course Name

Topic Name

© 2020 - Da Nang-Viet Nam

Delete Detail Page



Delete

Are you sure you want to delete this?

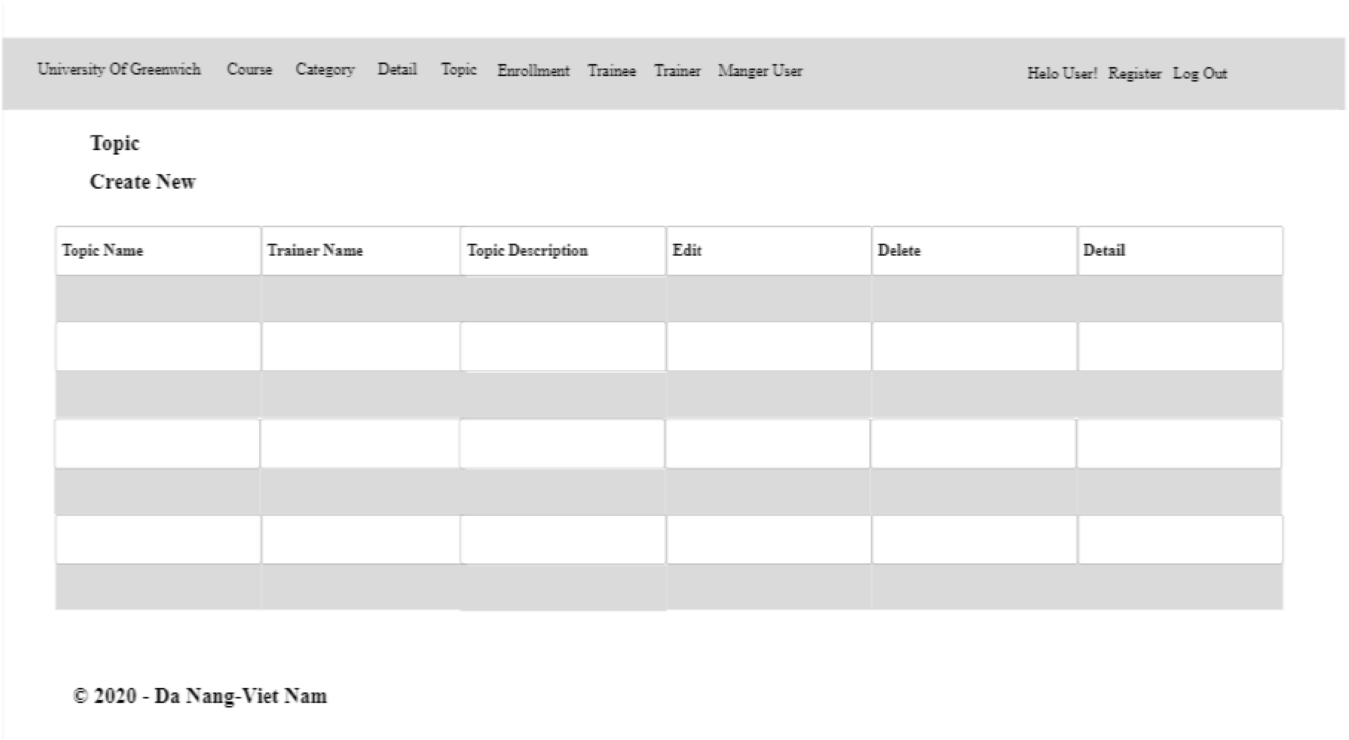
Detail

Course Name

Topic Name

© 2020 - Da Nang-Viet Nam

Topic Index Page



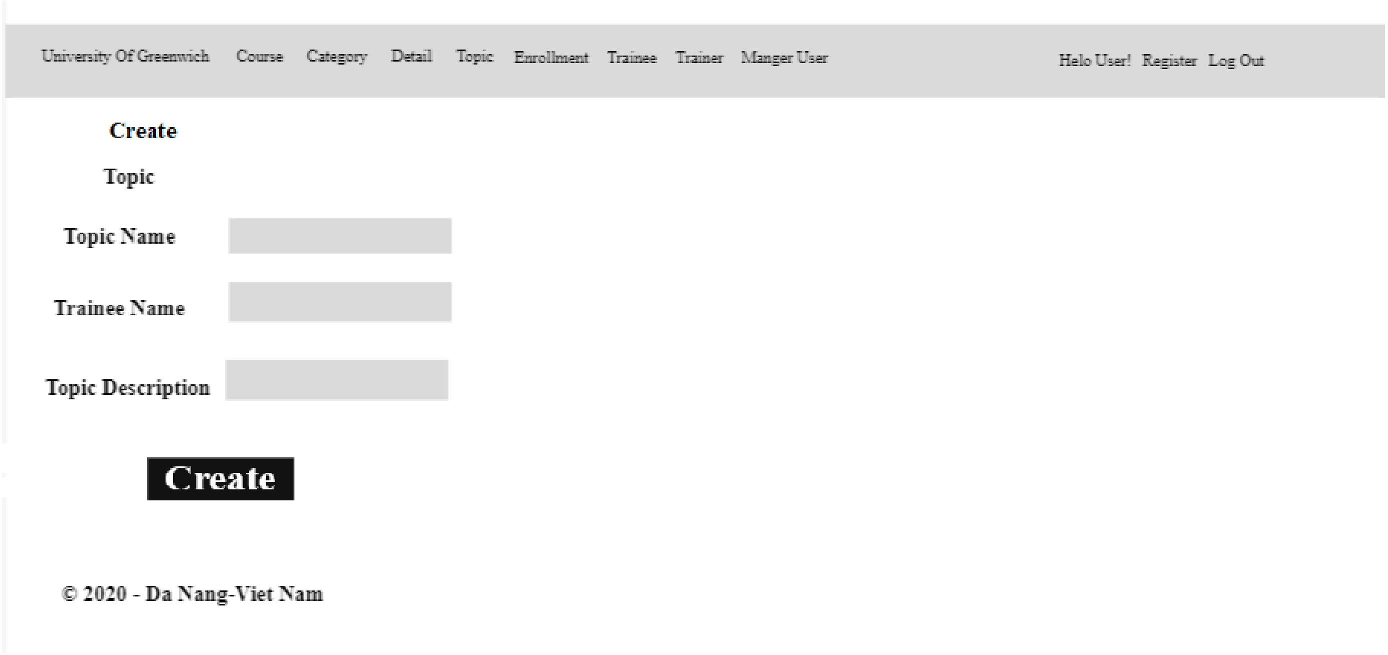
Topic

Create New

Topic Name	Trainer Name	Topic Description	Edit	Delete	Detail

© 2020 - Da Nang-Viet Nam

Create Topic Page



Create

Topic

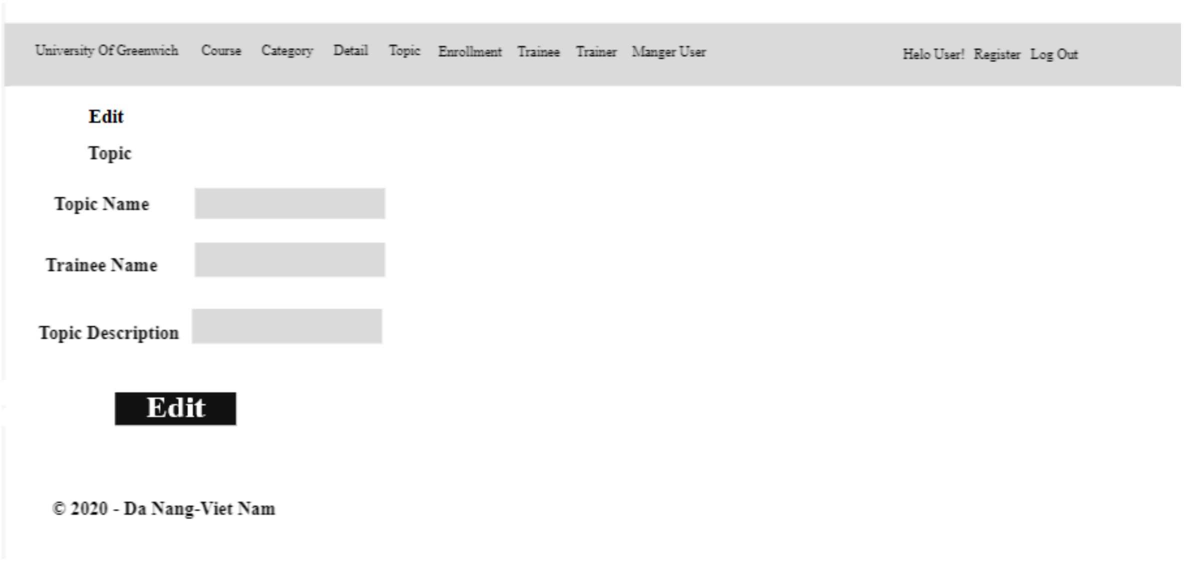
Topic Name

Trainee Name

Topic Description

© 2020 - Da Nang-Viet Nam

Edit Topic Page



Edit

Topic

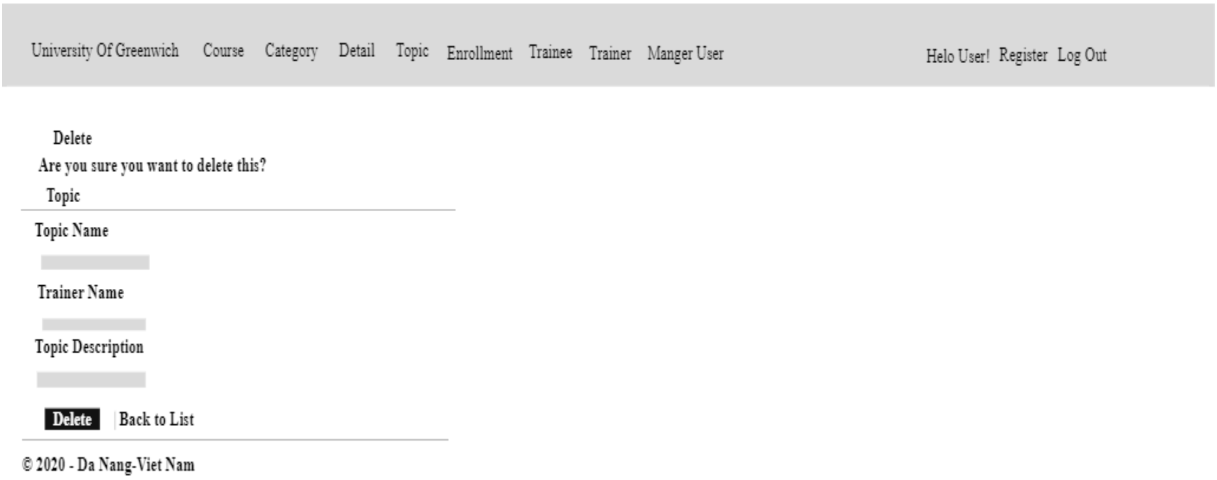
Topic Name

Trainee Name

Topic Description

© 2020 - Da Nang-Viet Nam

Delete Topic Page



Delete

Are you sure you want to delete this?

Topic

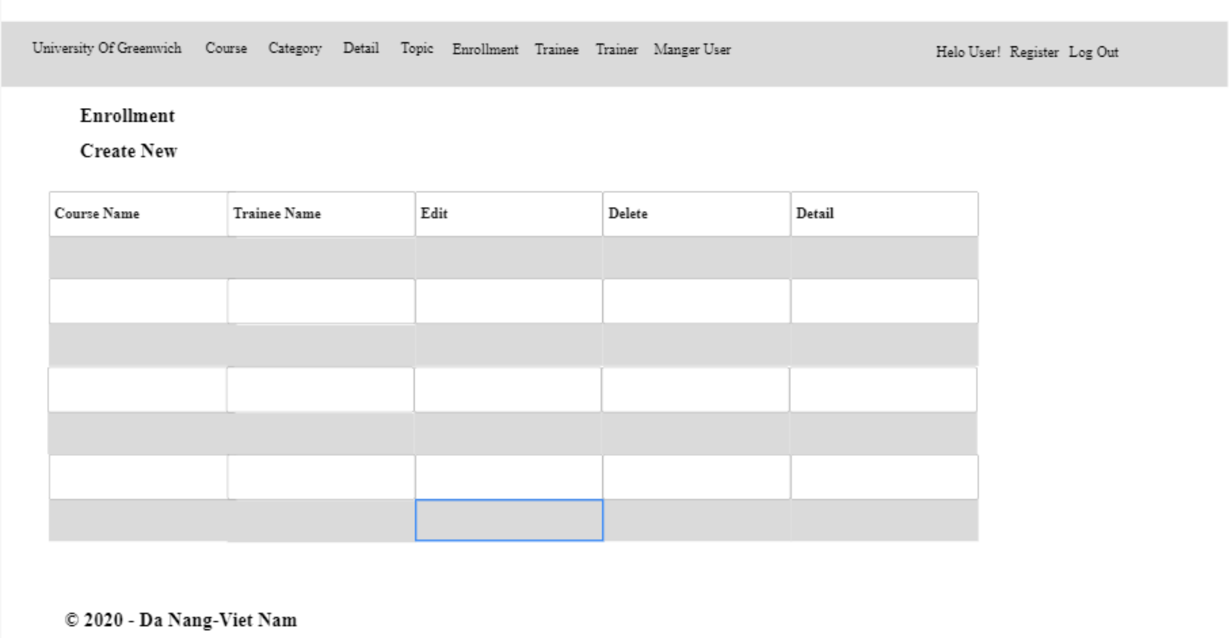
Topic Name

Trainer Name

Topic Description

© 2020 - Da Nang-Viet Nam

Enrollment Topic Index



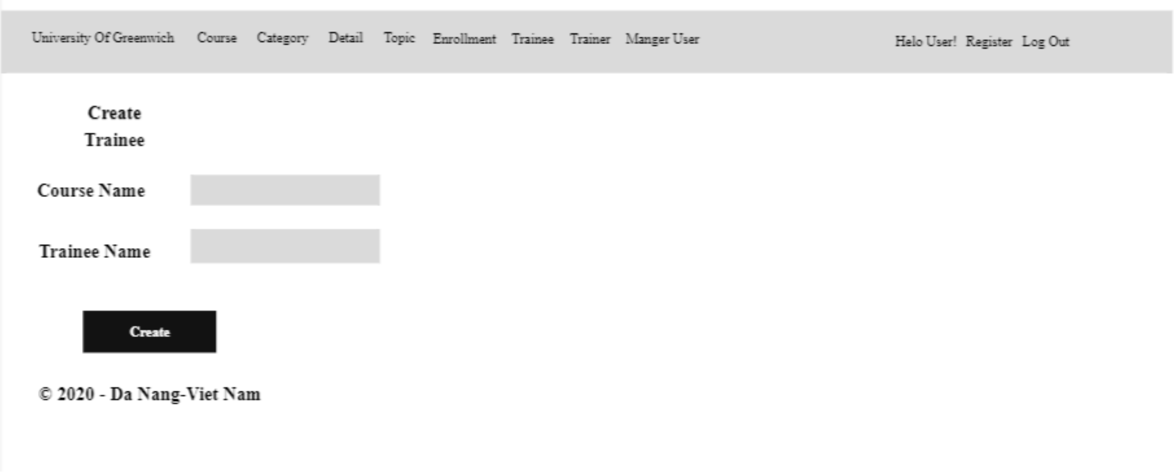
Enrollment

Create New

Course Name	Trainee Name	Edit	Delete	Detail

© 2020 - Da Nang-Viet Nam

Enrollment Create Page



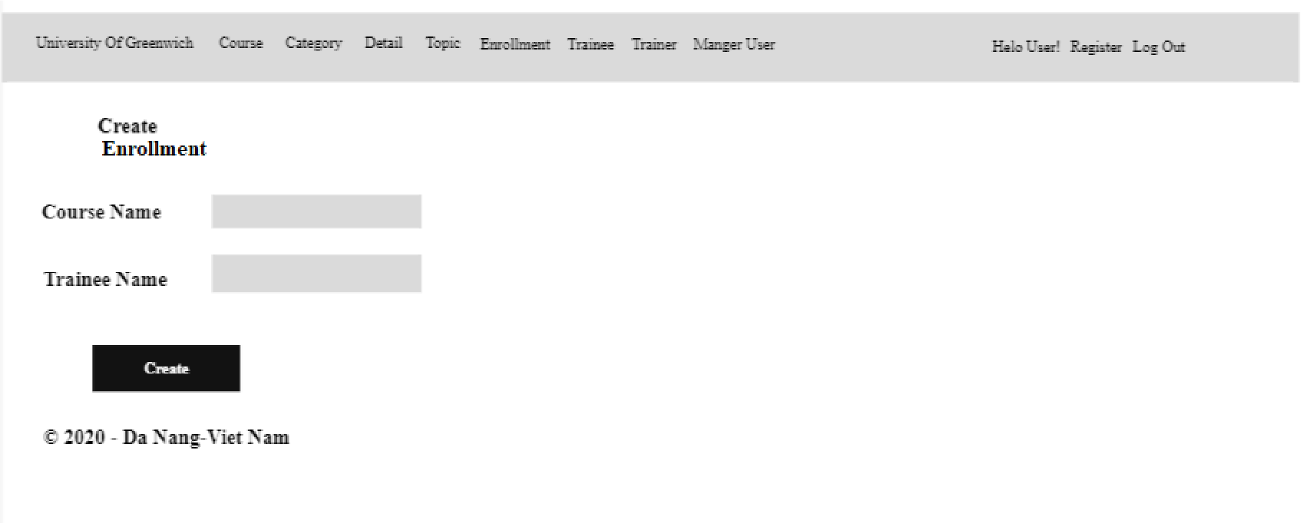
Create
Trainee

Course Name

Trainee Name

© 2020 - Da Nang-Viet Nam

Create Enrollment



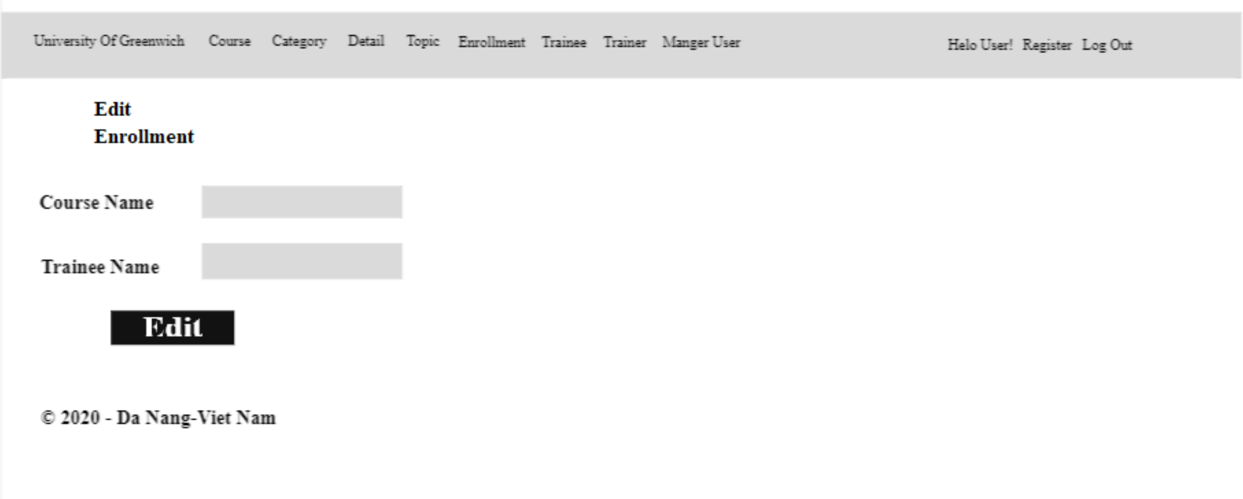
Create
Enrollment

Course Name

Trainee Name

© 2020 - Da Nang-Viet Nam

Edit Enrollment Page



University Of Greenwich Course Category Detail Topic Enrollment Trainee Trainer Manger User Helo User! Register Log Out

Edit
Enrollment

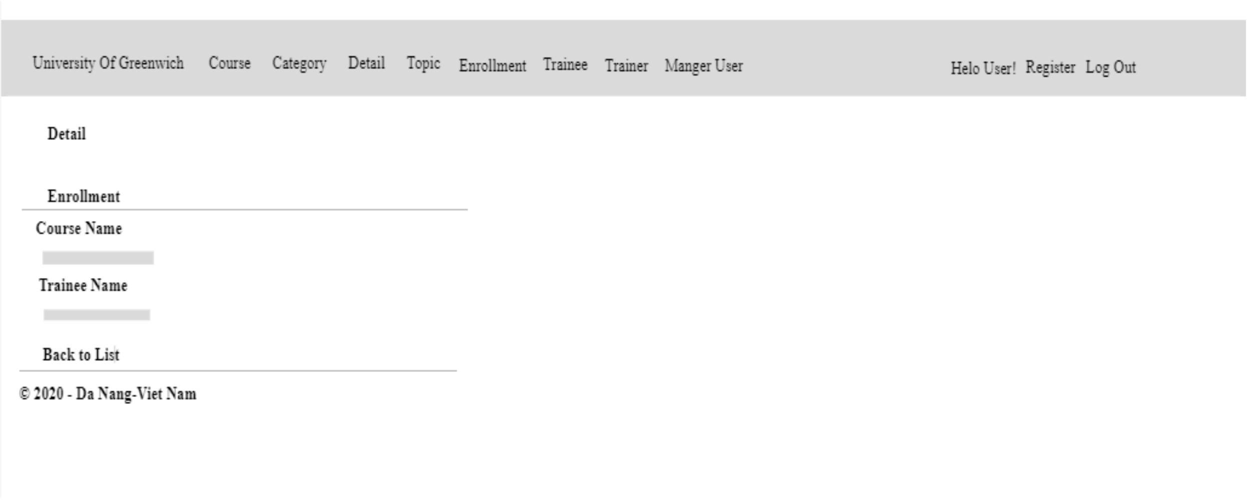
Course Name

Trainee Name

Edit

© 2020 - Da Nang-Viet Nam

Detail Enrollment Page



University Of Greenwich Course Category Detail Topic Enrollment Trainee Trainer Manger User Helo User! Register Log Out

Detail

Enrollment

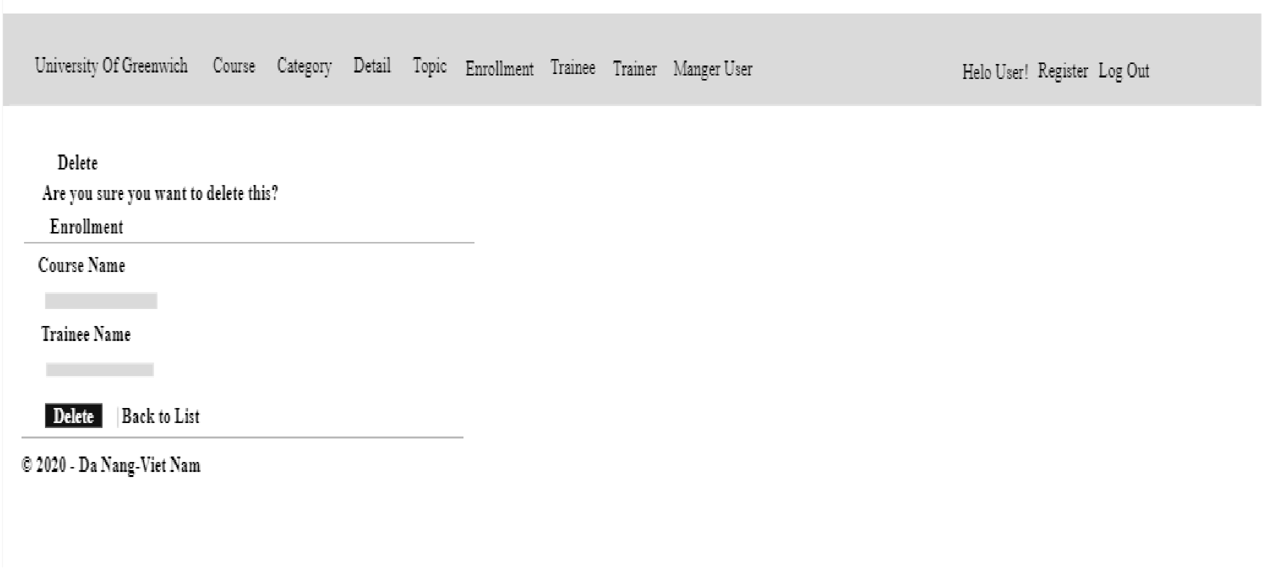
Course Name

Trainee Name

Back to List

© 2020 - Da Nang-Viet Nam

Delete Enrollment Page



Delete

Are you sure you want to delete this?

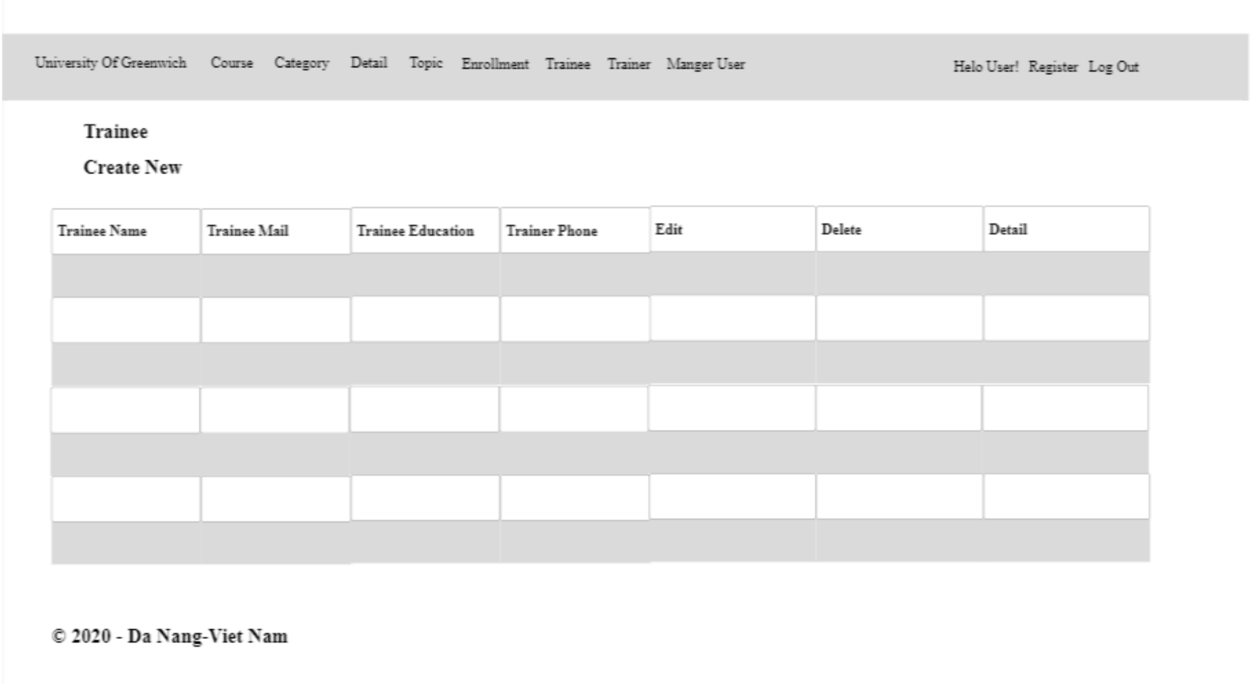
Enrollment

Course Name

Trainee Name

© 2020 - Da Nang-Viet Nam

Trainee Index Page

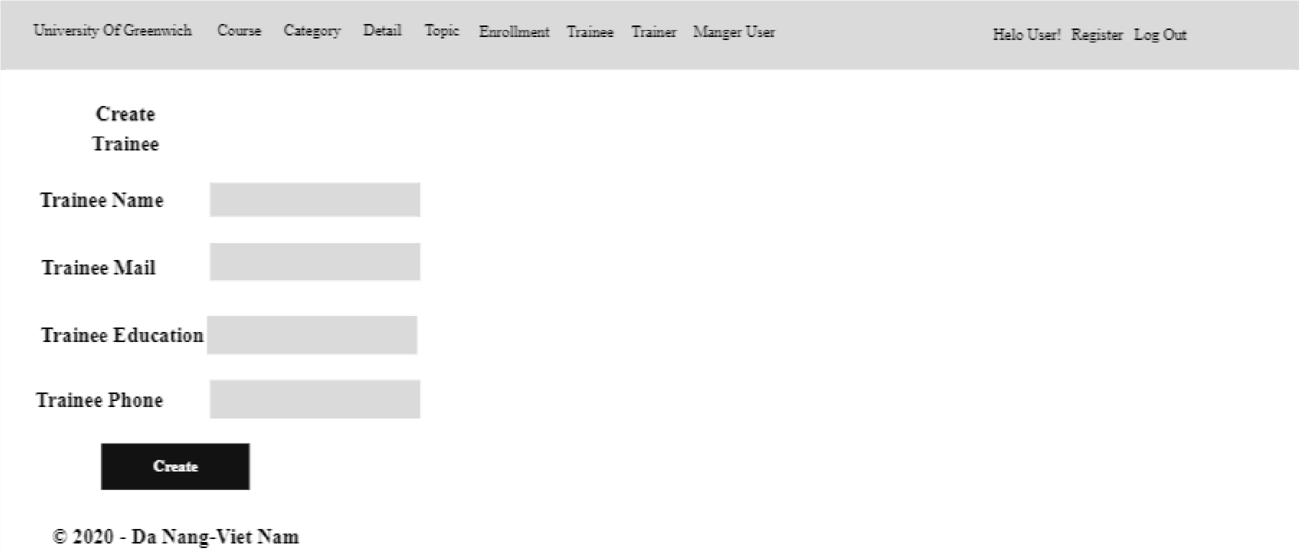


Trainee

Trainee Name	Trainee Mail	Trainee Education	Trainer Phone	Edit	Delete	Detail

© 2020 - Da Nang-Viet Nam

Trainee Create Page



Create Trainee

Trainee Name

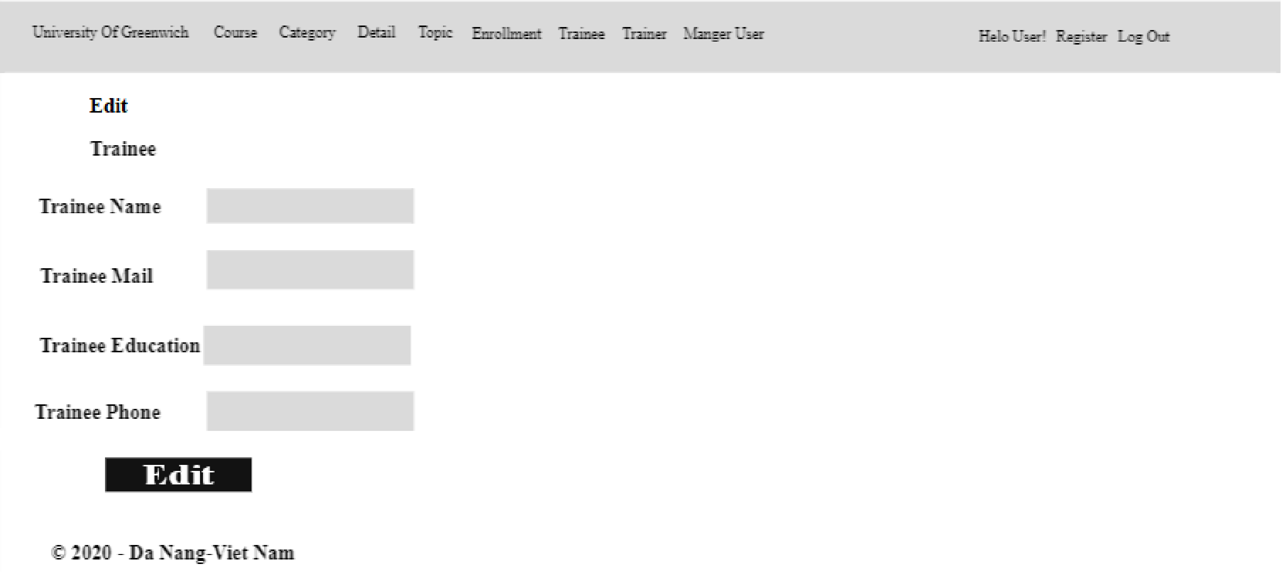
Trainee Mail

Trainee Education

Trainee Phone

© 2020 - Da Nang-Viet Nam

Trainee Edit Page



Edit Trainee

Trainee Name

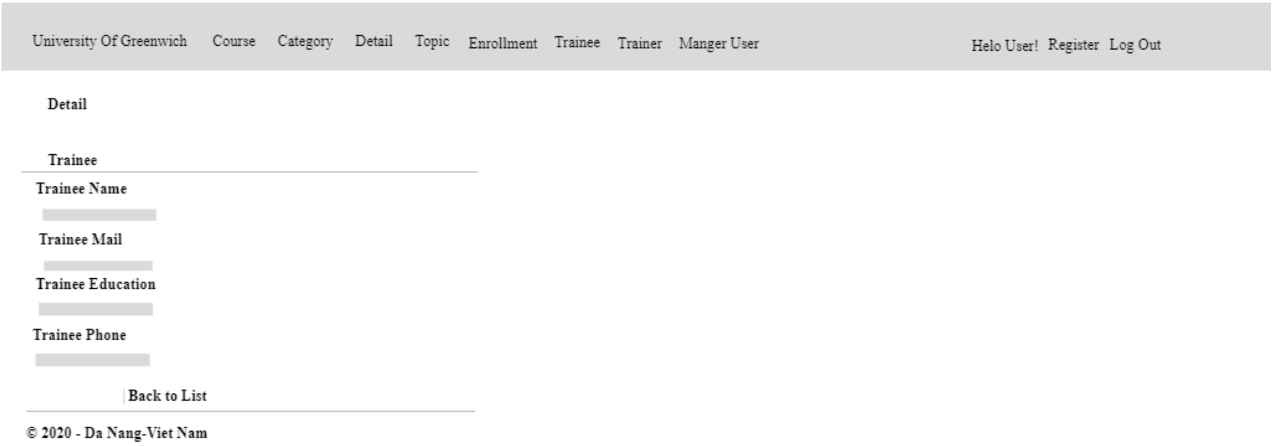
Trainee Mail

Trainee Education

Trainee Phone

© 2020 - Da Nang-Viet Nam

Trainee Detail Page



Detail

Trainee

Trainee Name

Trainee Mail

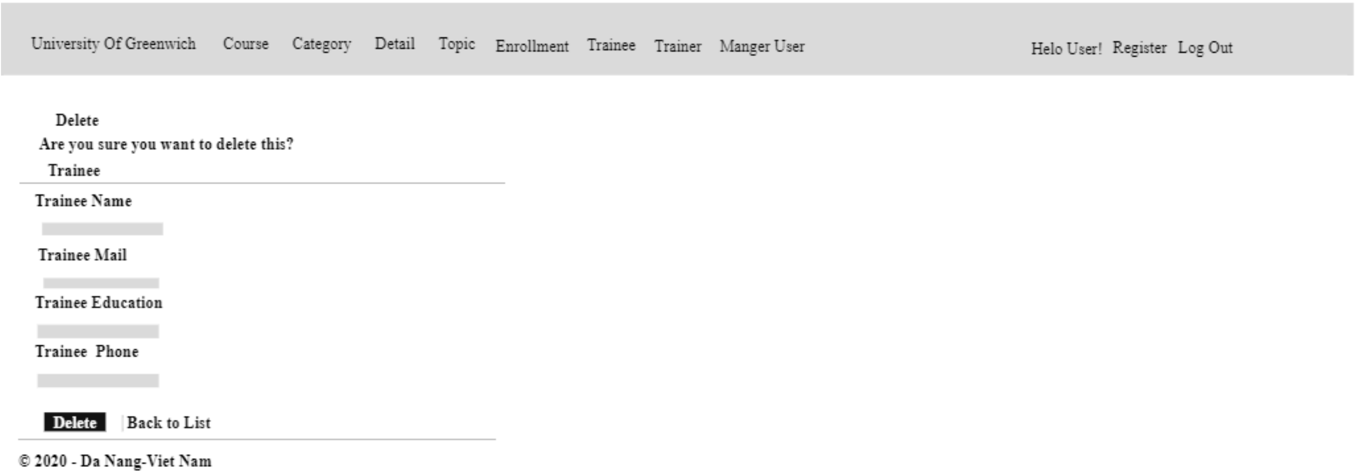
Trainee Education

Trainee Phone

|Back to List

© 2020 - Da Nang-Viet Nam

Trainee Delete Page



Delete

Are you sure you want to delete this?

Trainee

Trainee Name

Trainee Mail

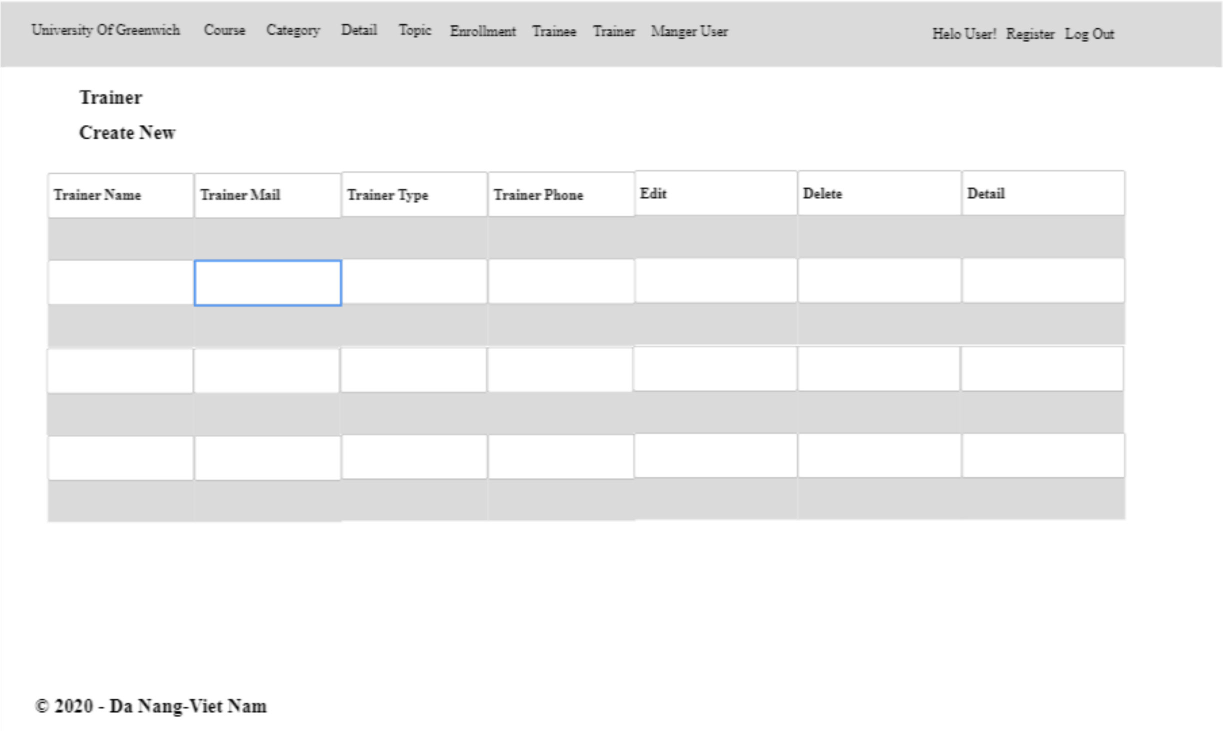
Trainee Education

Trainee Phone

|Back to List

© 2020 - Da Nang-Viet Nam

Trainer Index Page



University Of Greenwich Course Category Detail Topic Enrollment Trainee Trainer Manger User Helo User! Register Log Out

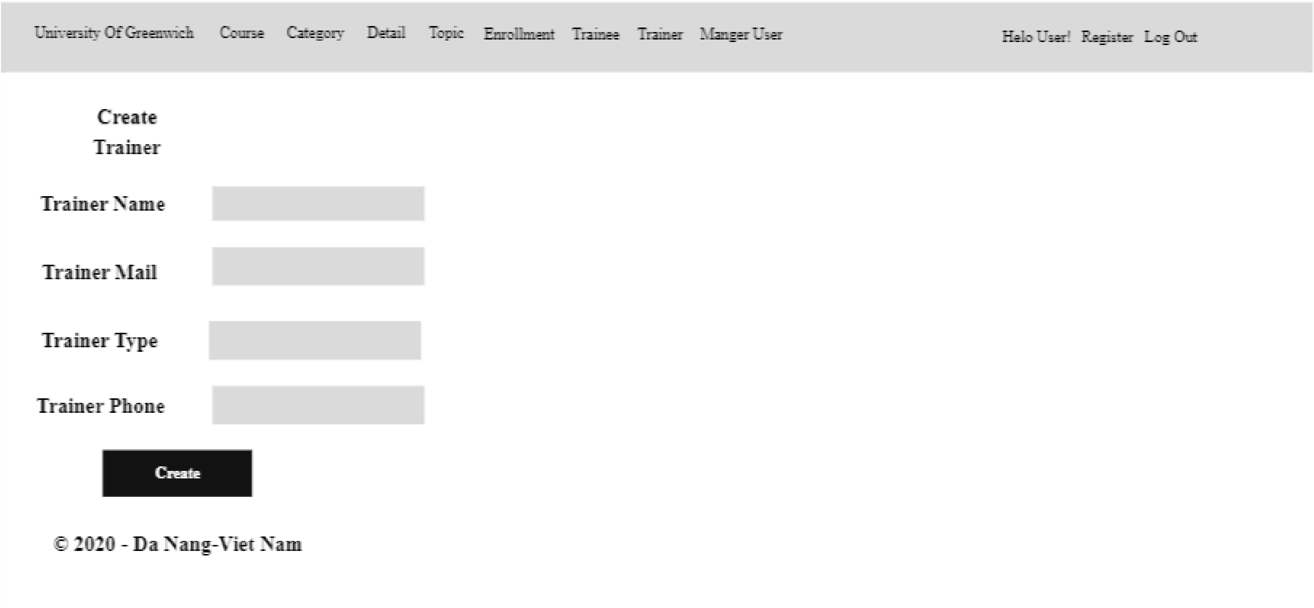
Trainer

Create New

Trainer Name	Trainer Mail	Trainer Type	Trainer Phone	Edit	Delete	Detail

© 2020 - Da Nang-Viet Nam

Trainer Create Page



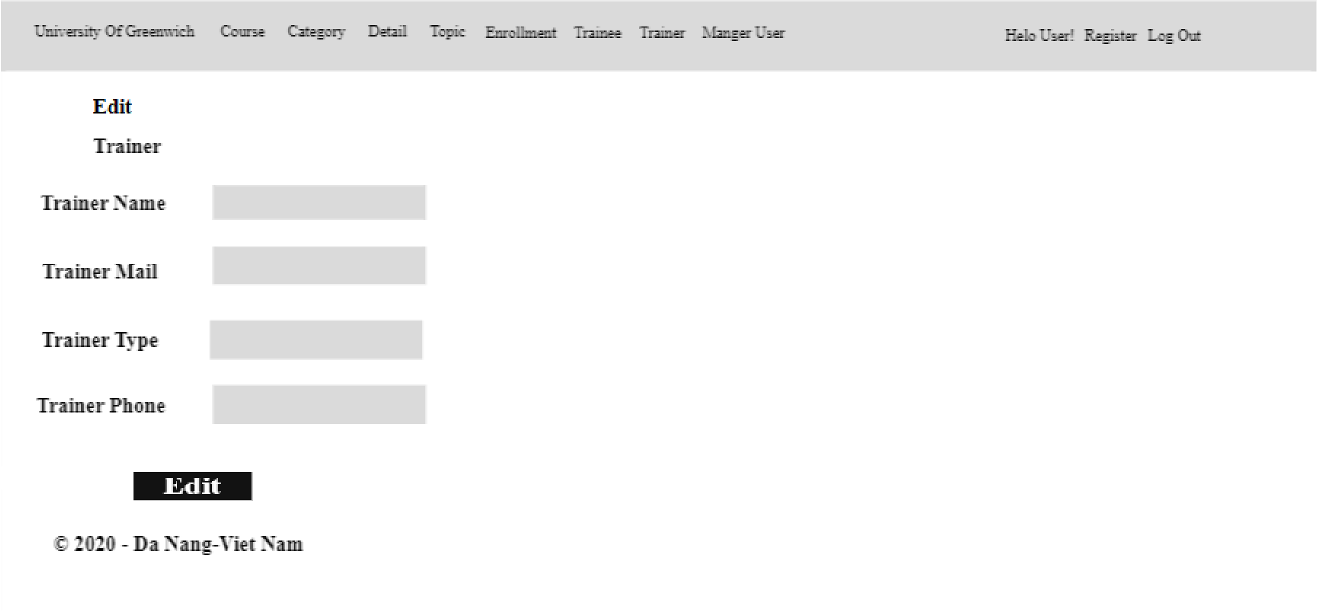
University Of Greenwich Course Category Detail Topic Enrollment Trainee Trainer Manger User Helo User! Register Log Out

Create
Trainer

Trainer Name	
Trainer Mail	
Trainer Type	
Trainer Phone	

© 2020 - Da Nang-Viet Nam

Trainer Edit Page



Edit

Trainer

Trainer Name

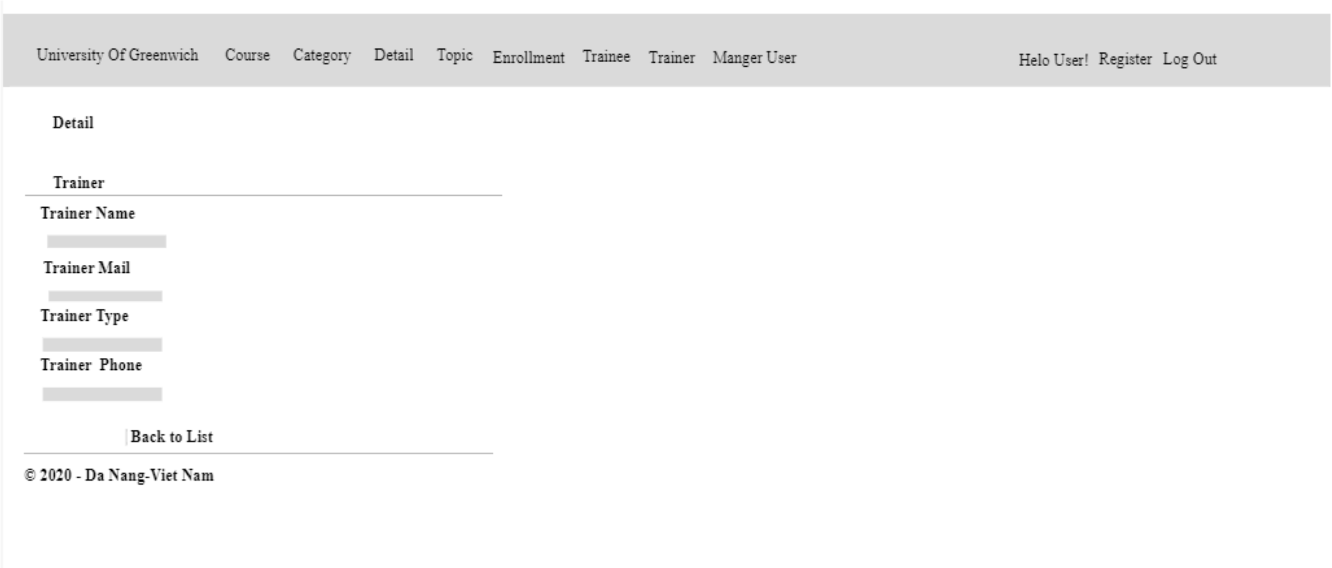
Trainer Mail

Trainer Type

Trainer Phone

© 2020 - Da Nang-Viet Nam

Trainer Detail Page



Detail

Trainer

Trainer Name

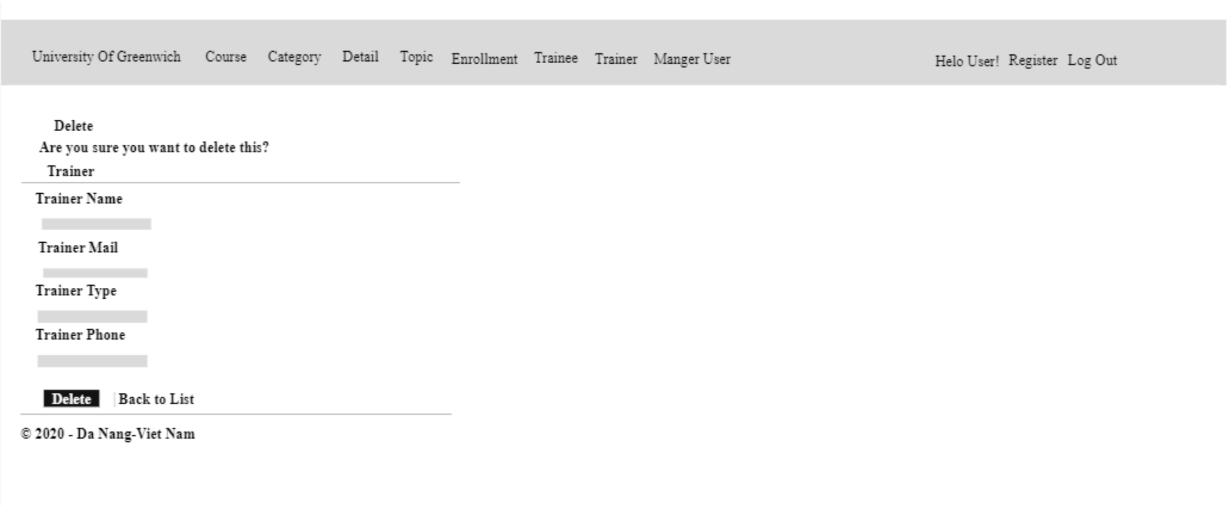
Trainer Mail

Trainer Type

Trainer Phone

© 2020 - Da Nang-Viet Nam

Trainer Delete Page



Delete

Are you sure you want to delete this?

Trainer

Trainer Name

Trainer Mail

Trainer Type

Trainer Phone

© 2020 - Da Nang-Viet Nam

5. NON-FUNCTIONAL REQUIREMENTS:

5.1. LIABILITY

- Required system database to be stored in Cloud
- System downtime for upgrade required less than 10 minutes
- Whenever performing upgrade, Index page The system is required to display an apology message
- system downtime for any reason (including upgrade, hacked) is no more than 1 day

5.2. ACCESSIBILITY

- The system is required to provide 2 languages: Vietnamese (VI) and English (EN)
- The user interface of the system must be responsive on many devices, including desktop, mobile and tablet devices

5.3. RELIABILITY

- System is required to supply up to 1 month for MTBF (Average Time Between Failures)
- The system is required to supply up to 2 times for NFUT (Number of errors per unit time)

5.4. SAFETY

- The system is required to only store User information for internal circulation

6. Rick of The Project

No.	Risk	Probability/Likelihood	Consequences	Preventive measures / minimizing the negative consequences
1	Database error	High	We chose to use the Database First method to make it easier for users to manage their database. But if they do not have a	Creating a new database and re- launching the website is very easy and fast, but if this happens, it can cause loss of customer database

			good database management, the database can be modified incorrectly and lead to software errors	
2	We lack access for customers especially Administrator access.	Low	We will provide users with an admin account and from here they can create the roles they want so they can manage and delegate the staff to the most accurate account.	Create a brochure for customers and on the first day of software delivery guide them carefully.
3	The content of the customer given in the text is incomplete and is urgently added in the final stage of completion.	High (Because customers are often busy to be able to identify the elements they most need or this is their first website and they have many new thoughts for it during execution.)	The project will not be completed properly in time; A delay, omission or late-stage update can take a long time to finalize a project on time if it is a large function. (Small)	We will meet and will provide customers with a specific questionnaire with what they give to understand each request as possible.
4	Customers do not know how they need a website	Medium(This may be their first website or the company's questionnaire creator, which clearly understands the requirements and presents the questionnaire incorrectly.)	The long delays, a lot of changes, costs more, stimulating both sides. (Short)	Divide the work correctly in the contract; take out the section about additional payments for major project changes, starting with the most needed feature. We should also spend more time talking to be able to properly capture customers' needs and budgets**.**
5	Report customers do not want to continue the project in the middle or near the completion of the project, or they look for a partner to lower the project.	Low	Wasted time and cost. (High)	Provide contractual compensation terms if either party automatically withdraws from the project or receives 40 or 50/100% of the amount agreed by both parties before the project starts to minimize the risk.

7. SOFTWARE DEVELOPMENT TOOLS AND TECHNIQUES SELECTION:

Teachnique:

Mvc Models:

MVC (Model-View-View-Controller) is a software architecture structure to create a user interface on the computer. MVC divided an application to three parts that can interact to separate information from the inside and information provided by the user.

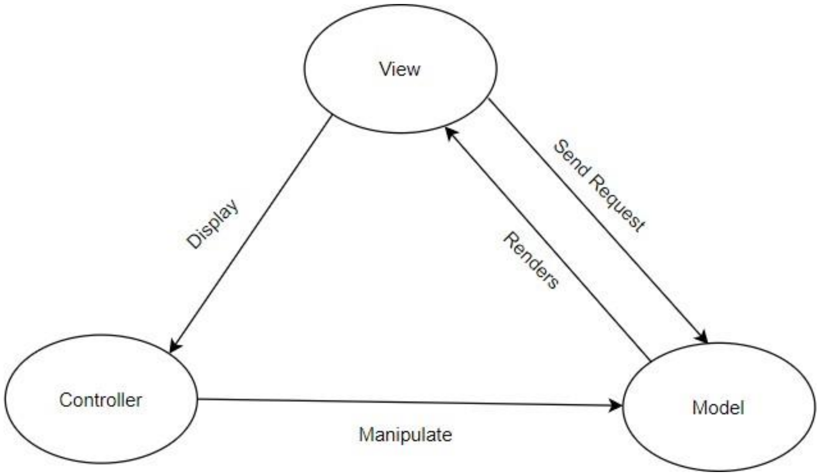
When it's used properly, the MVC prototype helps develop software developers to separate business protocols and interface-user interface. The software is developed by MVC prototype creates preservation because of specialized policies and highly specialized departments.

A pattern is a form of data and logic. It contains applications data. Restore and save the model status in the database to the subject of the model.

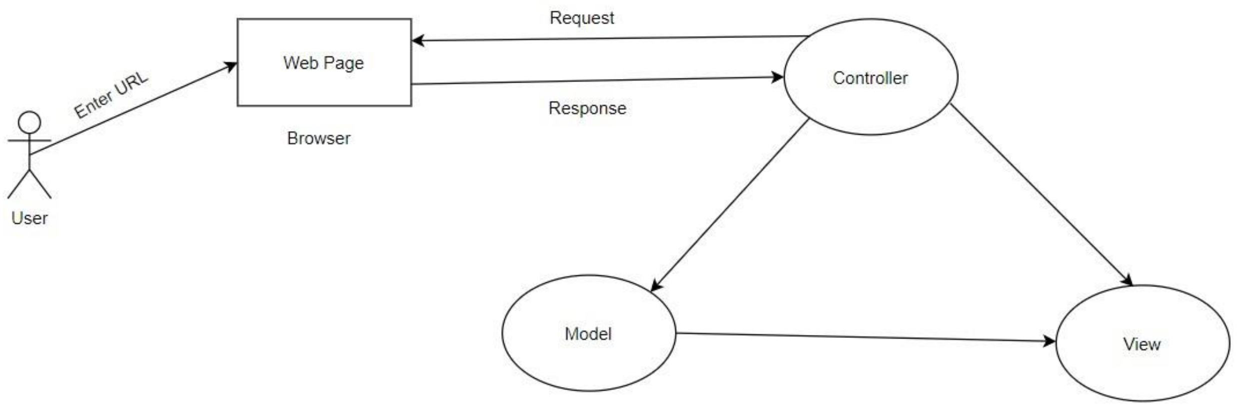
Consider it the user interface. Show the user-type data shows and allows them to change the data.

The controls are a requirement. The user usually communicates with the view mode, turns out the exact requests for the URL, the controls have to handle this request. With the stat as navigation as a guide, the controls to create the view.

The interaction between Model, View and Controller.



- illustrates the flow of the user's request in ASP.NET MVC.



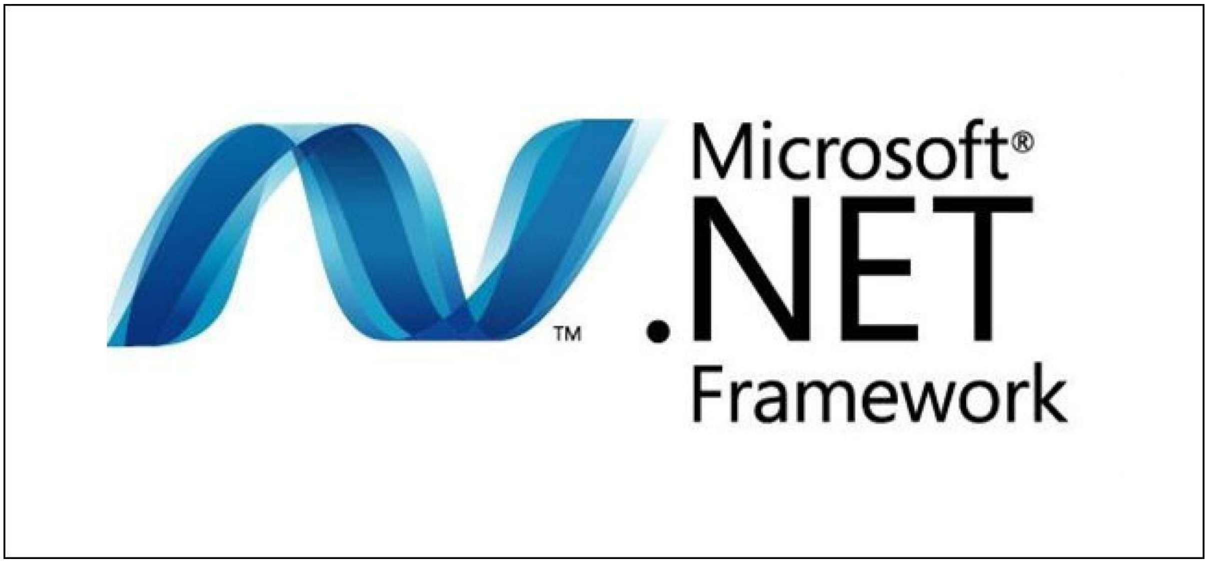
When the user enters a URL in the browser, as shown in the image above, it will come to the server and call the appropriate controller. The controller then uses the appropriate View and Model

Advantages and Disadvantages of MVC

Advantages	Disadvantages
- A key advantage of MVC is the separation of concerns. Separation of concerns means we divide the Application Model, Control, and View. - We can easily maintain our application because of interest. - At the same time, we can split multiple developers working at the same time. It will not affect one developer job to another developer job. - It supports TTD (test-based development). We can create an application with unit tests. We can write test case wins. - The latest version of MVC Support default responsive web and mobile templates. - We can create our own viewing tool. It is a very easy syntax compared to the traditional viewer.	- Cannot view design page preview like .aspx page. Every time you want to run, see the design. - Understanding the flow of the application is very difficult. It is a little complicated to implement and not suitable for small level applications. - It is a somewhat difficult implementation.

ASP.NET Framework:

ASP.NET is an open-source Web application platform designed to create dynamic Web pages created by Microsoft to enable programmers to create dynamic websites, applications, and services.



SQL Server:

Microsoft SQL Server is Microsoft database management system. It's a software product, like a database server, which is essentially stored and pursued data like other software applications for applications. You can run on the same machine or another network (including the Internet).

Microsoft Visual Studio

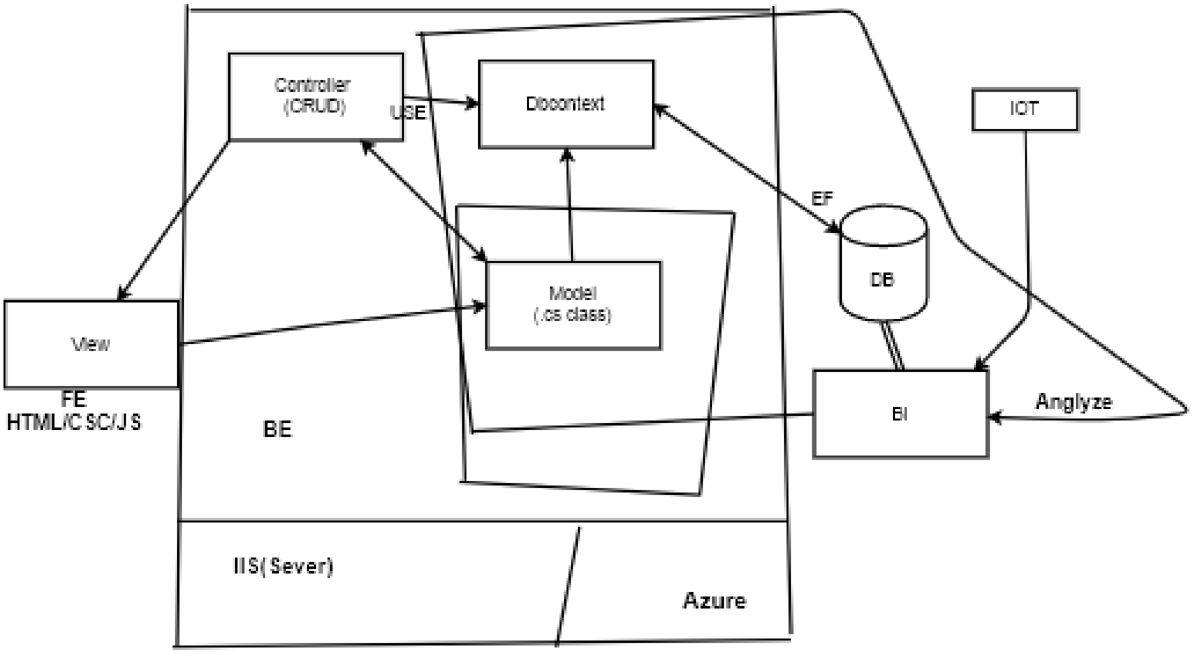
I’m choice Visual Studio because the ASP.NET Framework and SQL Server have Microsoft and I think that has best performant with Visual Studio and Microsoft Visual Studio is an Integrated Windows Development Environment (IDE). It is used for the development of Microsoft Windows computer programs as well as websites, web applications and web services. Visual Studio uses platforms such as Windows API, Windows Forms, Windows Presentation Foundation, Windows Store, and Microsoft Silverlight to develop software for Microsoft. It can create both the language of the computer and the code of management.

Azure:

I choose Azure because Azure is a collection of cloud services run by Microsoft and I use Visual Studio and SQL server, so Azure will be the cloud service providing the best compatibility. With Azure all infrastructure and geographic limitations are broken, you can build and deploy your tech ideas wherever you want, with a wide variety of tools, applications, and platforms that Azure can support you.

Azure offers over 500 different services, allowing you to build, deploy, and manage virtually any IT solution. It can be said, Azure is a world of unlimited possibilities. Whether you are a large business spanning multiple continents that need to run server workloads or a small business that wants a website with a global presence, Windows Azure can provide a platform. a platform for building applications that utilize the cloud to meet your business needs.

Activity diagram of azure with mvc



Tools

Draw.io

Draw.io is an open-source technology stack for building diagramming applications, and the world's most widely used browser-based end-user diagramming applications.

To draw diagrams: SRD, User case Diagram, Data flow diagram

Balsamiq.Cloud:

A web-based user interface design tool for creating wireframes (sometimes called mockups or low-fidelity prototypes).

To Draw mockup, screen my Application development.

GitHub:

GitHub is a system and version code management project that acts like a social network for developers. The setup members can replicate the source code from a repository and GitHub is a service server repository, each of which can create accounts on which to create their own repositories to have. can work.

I chose GitHub as the place to post my code because in Visual Studio we have integrated Git to directly push our code on GitHub without using a 3rd party or doing complicated steps.

8. COMPARISON BETWEEN TOOLS/TECHNIQUES AND JUSTIFICATION

	Wins	**Xampp**	**Mean**	**Mern**
What is this	is the NetBIOS Name Service (NBNS) introduced by Microsoft, a name server and service for NetBIOS device names? Yes, WINS is to NetBIOS names what DNS is to domain names — a central mapping of hostnames to	is a free and open-source cross-platform web server solution stack package developed by Apache Friends, consisting mainly of the Apache HTTP Server, MariaDB database, and interpreters for scripts written in the PHP and Perl	a free and open-source JavaScript software stack for building dynamic web sites and web applications.	is a web development framework. It consists of MongoDB, ExpressJS, ReactJS, and NodeJS as its working components.

	network addresses.	programming languages		
Database	SQL Server	MariaDB or MySQL	MongoDB	MongoDB
Programing language	C#	Perl	JavaScript	JavaScript
Web Server	Internet Information Services	Apache	Node.js	Node.js
Operating system	Windows Server	Cross-flatform	Cross-flatform	Cross-flatform
Database	SQL Server	MariaDB or MySQL	MongoDB	MongoDB

We choose WINS to develop my application, because I'm using Windows Serves Operation System

Link Code:

https://github.com/trunganh12399/WebUoFASM1

Referent:

https://www.guru99.com/mvc-tutorial.html
https://en.m.wikipedia.org/wiki/Solution_stack
http://www.discoversdk.com/blog/which-full-stack-developer-are-you
https://www.quora.com/What-is-the-use-of-XAMPPhttps://studyreadeducate.blogspot.com/2017/03/introduction-of-xampp-lamp-wamp.htm